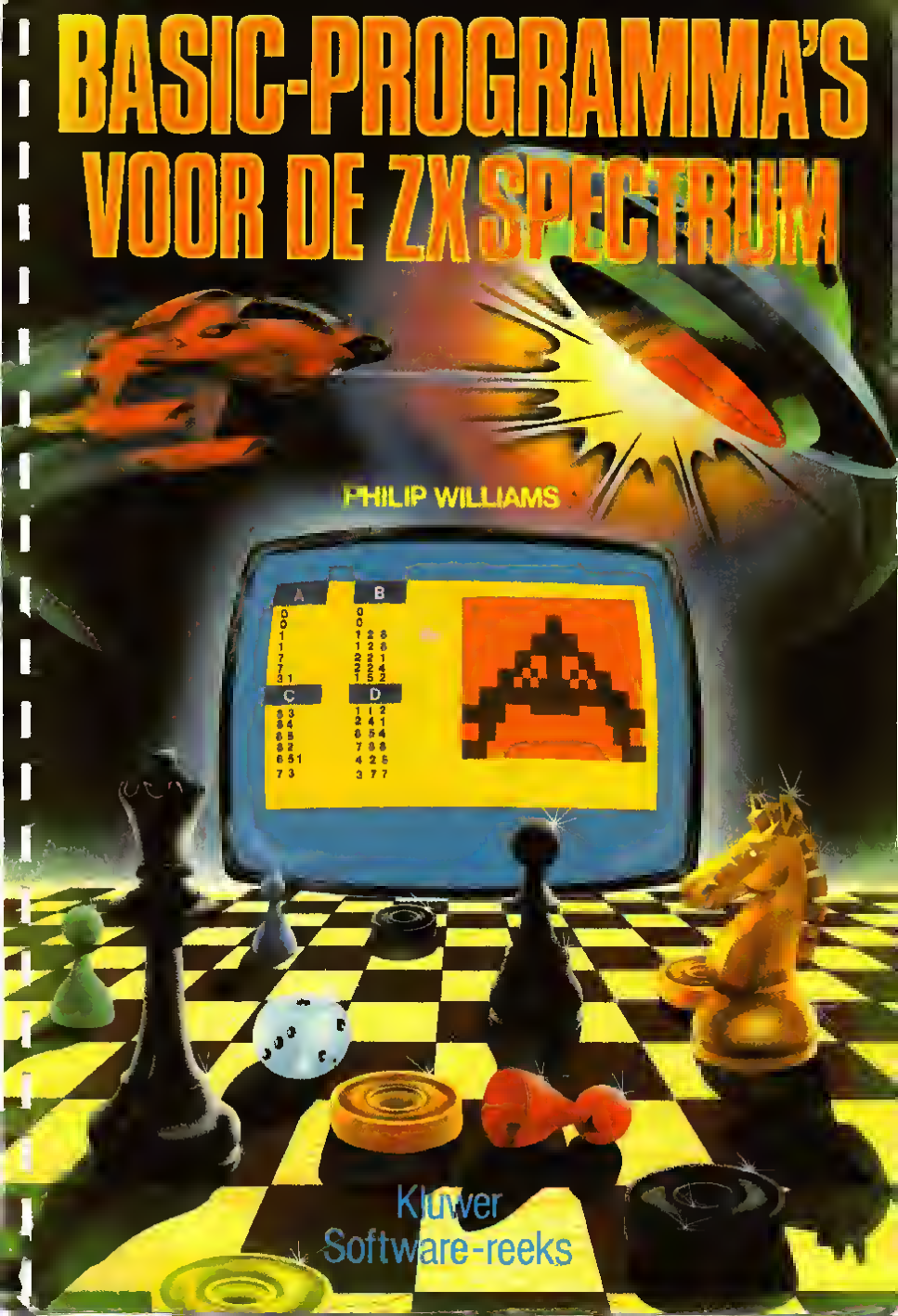
BASIC-PROGRAMMA'S VOOR DE ZX SPECTRUM

PHILIP WILLIAMS



Kluwer
Software-reeks

BASIC-programma's voor de ZX Spectrum

Philip Williams

BASIC-programma's voor de ZX Spectrum

Kluwer Technische Boeken B.V.
Deventer - Antwerpen

Vertaald door Marcel Brugmans

ISBN 90 201 1718 1
D/1984/0108/171

Oorspronkelijke titel: Over the Spectrum
Uitgegeven bij: Melbourne House Publishers

© 1982 Melbourne House (Publishers) Ltd
© 1984 van de Nederlandse vertaling bij
Kluwer Technische Boeken B.V. - Deventer

1e druk 1984

Niets uit deze uitgave mag worden verveelvuldigd en of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.

Ondanks alle aan de samenstelling van de tekst bestede zorg, kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

Inhoud

Over de programma-listings 7

1. Algemene programma's 8
1.1 Springkikker 8
1.2 Getalspiegel 10
1.3 Asteroïden in de ruimte 12
1.4 Spectrumklok 14
1.5 3-D Doolhof 18

2. Onderwijskundige programma's 22
2.1 Meetkundige test 22
2.2 Het Huis van Oranje 23

3. Gokspelletjes 26
3.1 Blackjack 26
3.2 Fruitmachine 30

4. Wiskundige programma's 37
4.1 'Bubble'-sorteerroutine 37
4.2 Stelsel van vergelijkingen 41

5. Arcade-spelletjes 44
5.1 Vlucht uit de ruimte 44
5.2 Maanlander 47
5.3 Alien blitz 51
5.4 Spectrum-invaders 54
5.5 Meteorenstorm 58
5.6 Kidnap 64
5.7 Snelwegkikker 69

6. Hulpprogramma's 80
6.1 Vrij programmeerbare karakters 80
6.2 Regelhernummering 86
6.3 Regelblokverwijder 91
6.4 Machinetaalmonitor 93

7. Zakelijke programma's 99
7.1 Loonlijst 99
7.2 Verkoopanalyse 105
7.3 Bezittingenevaluatie 109

8. Strategische programma's 114
8.1 Schaken 114
8.2 Dammen 118
8.3 Avontuur 128

Over de programma-listings

Zoals u ongetwijfeld weet zijn er verschillende 'modes' beschikbaar op de Spectrum. Deze geven elk verschillende 'keywords' of grafische symbolen.
De programma-listings in dit boek zijn zo ontworpen dat zij zo gemakkelijk mogelijk leesbaar zijn. Enkele gebruikte symbolen behoeven enige uitleg.

\# Dit symbool geeft aan, dat er een spatie ingetikt moet worden in een PRINT-statement of in een STRING.
Spaties zijn in deze programma's toegevoegd om de leesbaarheid te bevorderen. Soms zijn zij essentieel voor de goede werking van het programma.

SHI Dit simbool geeft aan dat de CAPS SHIFT toets ingedrukt gehouden moet worden voor het volgende karakter.

EXT Dit symbool wordt gebruikt om aan te geven dat u de Spectrum in EXTENDED MODE moet zetten. Hiertoe moet u CAPS SHIFT en de SYMBOL SHIFT toetsen tegelijk indrukken. Zie de Spectrum-handleiding voor meer details.

GRA Hiermee geven wij aan, dat u de Spectrum in GRAPHICS MODE moet zetten. Dit doet u door, terwijl u de toets CAPS SHIFT ingedrukt houdt, de '9'-toets in te drukken.

GRU Dit symbool (GRAPHICS uit) gebruiken wij om aan te geven, dat u GRAPHICS MODE uit moet zetten (druk, terwijl u de CAPS SHIFT toets ingedrukt houdt, de 9-toets in). Merk op dat GRAPHICS MODE ook uitgezet wordt door *EXT*.

INV Dit geeft INVERSE VIDEO MODE aan. Druk de toetsen CAPS SHIFT en '4' in.

TRU Dit geeft TRUE VIDEO MODE aan. Druk de toetsen CAPS SHIFT en '3' in.

In dit boek wordt gebruik gemaakt van de mogelijkheid van de Spectrum om papier- en inktkleuren aan te geven in een PRINT-statement. Dit wordt gedaan door EXTENDED MODE aan te zetten en de juiste kleurtoets (voor papierkleur) of CAPS SHIFT + de juiste kleurtoets (voor inktkleur) in te drukken.
In het boek is dit bijvoorbeeld op de volgende manier aangegeven:

EXT SHI 3	kies inktkleur 3 (magenta)
EXT 6	kies papierkleur 6 (geel)

Bij de bovengenoemde voorbeelden moet u als volgt te werk gaan:

EXT SHI 3
1. Druk de toetsen CAPS SHIFT en SYMBOL SHIFT tegelijk in. De cursor verandert in een 'E'.
2. Houd de CAPS SHIFT toets ingedrukt en druk de '3' in. De cursor verandert weer in een 'K' of 'L'.

EXT 6
1. Druk de toetsen CAPS SHIFT en SYMBOL SHIFT tegelijk in. De cursor verandert in een 'E'.
2. Druk de '6' in. De cursor verandert weer in een 'K' of 'L'.

1. Algemene programma's

1.1 Springkikker
(Copyright Beam Software)

Het spelletje springkikker is aardig en simpel U begint met twee zich tegenover elkaar bevindende groepen kikkers, en elke kikker mag alleen maar naar een naastliggende plaats verhuizen of over een andere kikker heenspringen.

```
X X X X - O O O O
1 2 3 4 5 6 7 8 9
```

Dus om te beginnen mag bijvoorbeeld de kikker op de plaats 4 naar plaats 5 gaan, of de kikker van 6 naar 5, of de kikker van 3 naar 4 springen zodat hij op 5 uitkomt, of die van 7 springt over 6 naar 5.
Het doel van het spel is om alle linker kikkers rechts te krijgen en omgekeerd echter in het minste aantal beurten. Probeert u het maar.

Programma-opbouw
In het eerste gedeelte van elk programma moeten de variabelen, die eventueel gebruikt worden, benoemd worden. In dit geval wilt u de beginpositie van de kikkers definiëren en het aantal tot nu gebruikte beurten op nul zetten.
Een bestudering van het programma geeft de volgende opbouw.
Benoem de variabelen

Print	print positie van kikkers
	test of het einde van het spel bereikt is,
	zo ja ga naar Einde
Invoer	voer de spelerzet in
	controleer of de zet legaal is,
	zo niet, ga naar Invoer
	vermeerder het aantal beurten met 1
	voer de zet uit en ga naar Print
Einde	feliciteer speler
	vraag aan speler of hij nog een keer wil spelen, zo ja, geef RUN

Deze simpele 'top-bodem'-benadering geeft u een overzicht van de programma opbouw, zodat u het programma beter begrijpt en er later zonodig veranderingen in kunt aanbrengen.

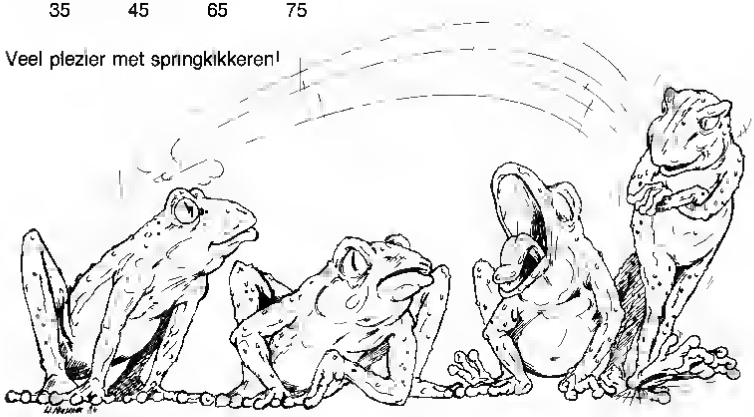
Opbouw van de variabelen
In dit programma gebruikt u 'string'-variabelen om de positie van de kikkers vast te kunnen leggen. Een 'string'-variabele is hier gemakkelijk te hanteren en het versnelt het op het scherm zetten van de 'string'.
U definieert a$ als de oorspronkelijke positie van de kikkers en b$ als de huidige positie. U kunt dezelfde variabelen gebruiken om te controleren of u al klaar bent (zie regel 150).

U gebruikt de variabelen 'naar' en 'van' om de positie aan te geven waarnaar en waarvan de kikker springt. Omdat een kikker alleen maar naar een lege positie mag springen, kunt u dit gemakkelijk controleren (zie regel 200).
De rest van het programma is duidelijk, u gebruikt de 'teller' voor het aantal gebruikte beurten.

Gebruik van het programma
Het programma verwacht een getal van 2 cijfers als invoer om de sprongen 'van' en 'naar' te benoemen. Het programma zal, bijvoorbeeld als legale eerste zet, alleen een van de volgende getallen accepteren:

35 45 65 75

Veel plezier met springkikkeren!



Programma

```
  10 REM kikker
 100 LET a$='GRA SHI 8 # SHI 8 # SHI 8 # SHI 8 ### SHI 6 #
SHI 6 # SHI # 6 SHI 6 GRU'
 110 LET b$=a$
 120 LET teller=0
 130 CLS
 140 PRINT AT 5,3;b$;AT 6,3;"1#2#3#4#5#6#7#8#9"
 150 IF b$(1 TO 7)=a$(11 TO 17) AND b$(11 TO 17)=a$(1 TO 7)
THEN  GO TO 250
 160 INPUT "Je#zet#svp#";k$
 170 IF LEN k$<>2 THEN  GO TO 160
 180 LET van=2*(CODE k$(1)-48)-1
 190 LET naar=2*(CODE k$(2)-48)-1
 200 IF b$(naar)<>"#" OR ABS (naar-van)>4 THEN  GO TO 160
 210 LET teller=teller+1
 220 LET b$(naar)=b$(van)
 230 LET b$(van)="#"
 240 GO TO 140
 250 PRINT 'Je#deed#er#";teller;'#beurten#over'
 260 INPUT "Nog#een#keer?#";k$
 270 IF CODE k$=65 THEN  RUN
```

1.2 Getalspiegel
(Copyright Beam Software)

Dit is een spiegel om het logisch denken te testen! U begint met negen getallen, van 1 tot 9, in een willekeurige volgorde.
Het doel is om ze in zo min mogelijk beurten in de goede volgorde te krijgen. Het enige middel dat u heeft om de volgorde te veranderen is de mogelijkheid om een aantal getallen te spiegelen.
Als u bijvoorbeeld als eerste drie getallen '3 2 1' had, dan krijgt u '1 2 3' als u deze eerste drie spiegelt. Als u alleen de eerste twee gespiegeld zou hebben, dan had u '2 3 1' gekregen.

Let op dat u alleen kunt opgeven hoeveel getallen u gespiegeld wilt hebben. Dit programma is betrekkelijk kort, dus het is gemakkelijk in te voeren op uw computer, maar het is zeker een uitdaging voor het logisch denken.

Programma-opmerkingen
Dit programma benadrukt het gebruik van 'arrays', en laat tevens zien dat programma's die zinnige namen gebruiken voor variabelen gemakkelijk te ontcijferen zijn.
Bekijkt u de programma-uitdraai. Het is niet nodig om een aparte programma-opbouw te geven omdat alle informatie in de uitdraai staat.
Merk op dat het mogelijk is om spaties aan te brengen in het begin van een regel, zodat FOR-NEXT-lussen enz. extra benadrukt worden.

Het gebruik van namen als variabelen voor regelnummers in GOTO- en GOSUB-opdrachten is ook een goede manier om een overzicht te geven van de programma-opbouw. Het beperkt ook het werk dat u moet doen tot een minimum als u de HERNUMMER-routine, die verderop in het boek staat, gaat gebruiken.

Gebruik van arrays
Het gebruik van arrays maakt het mogelijk om naar een element te verwijzen zonder dat ieder element afzonderlijk een naam hoeft te hebben.
In het begin van het programma dimensioneren we de array en deze zet elk element op nul. De grootte van de array wordt bepaald door een variabele, zodat u alleen maar de waarde van deze variabele hoeft te veranderen als u het logisch denken aan de hand van een langere rij getallen wilt testen.

Bestudeer de manier waarop de getallen in een schijnbaar willekeurige volgorde worden gezet - we nemen een voor een de getallen en ruilen dat met het getal uit een willekeurig gekozen positie. Dit algoritme is ook een zeer bruikbare en efficiënte manier om kaarten in een kaartspel te schudden.

Programma

```
  10 REM spiegel
 100 LET print=600
 110 LET spiegel=400
 120 LET invoer=300
 130 LET grootte=9
 140 DIM a(grootte)
 150 FOR c=1 TO grootte
 160   LET a(c)=c
 170 NEXT c
 180 FOR c=grootte TO 2 STEP -1
 190   LET ruil=1+INT (grootte*RND)
 200   LET tijdelijk=a(c)
 210   LET a(c)=a(ruil)
 220   LET a(ruil)=tijdelijk
 230 NEXT c
 240 LET teller=0
 250 GO SUB print
 300 INPUT "Hoeveel#getallen#wil#je#########spiegelen?#";ho
eveel
 310 IF hoeveel=0 THEN  STOP
 320 IF hoeveel<=grootte THEN  GO TO spiegel
 330 PRINT grootte;'#is#het#maximum!"
 340 GO TO invoer
 400 PRINT "In#spiegel#";hoeveel;'#getallen"
 410 LET teller=teller+1
 420 FOR d=1 TO INT (hoeveel/2)
 430 LET tijdelijk=a(d)
 440 LET draai=hoeveel-d+1
 450 LET a(d)=a(draai)
 460 LET a(draai)=tijdelijk
 470 NEXT d
 480 GO SUB print
 490 FOR d=1 TO grootte
 500 IF a(d)<>d THEN  GO TO invoer
 510 NEXT d
 520 PRINT "Je#had#";teller;'#beurten#nodig."
 530 STOP
 600 POKE 23692,255
 610 FOR d=1 TO grootte
 620 PRINT a(d);"#";
 630 NEXT d
 640 PRINT
 650 RETURN
```

1.3 Asteroiden in de ruimte
(Copyright Neil Streeter)

U reist door de ruimte in uw ruimteschip als u plotseling in een ruimtestorm terechtkomt.
U kunt uw schip alleen maar door het puin sturen door het roer te gebruiken (toets '5' naar links, en toets '8' naar rechts) - de voortstuwing is door een van de meteoren uitgeschakeld.
Alsof dit nog niet erg genoeg is zult u merken dat als u de meteorenstorm heeft overleefd u zo de weg kwijt bent dat u de verkeerde kant op reist in de ruimtebanen. Al het ruimteverkeer komt recht op u af.
U moet dit verkeer op dezelfde manier ontwijken als hiervoor, maar dit is moeilijker omdat de ruimteschepen groter zijn dan het meteorenpuin.
Uiteindelijk zult u crashen ! U vindt dan uw overlevingsscore op het scherm van uw ruimteschipconsole.

Programma-opbouw
Dit programma bootst de SCROLL-functie na, die op de meeste computers aanwezig is, maar niet op de Spectrum.
Hetzelfde resultaat kan echter worden verkregen door de waarde van de variabele SCRCT te veranderen (dit gebeurt door een POKE naar het adres 23692 - zie Machinetaalmonitor-programma voor meer informatie over PEEK en POKE), en dan twee spaties te printen op het einde van het scherm (bijv. AT 21,31).

Als het scherm een SCROLL krijgt dan zal het schip samen met de rest verschuiven. Door het schip nu met spaties te overschrijven en het dan weer op de juiste positie neer te zetten blijft het schip altijd op dezelfde regel van het scherm terwijl de rest beweegt.
Het meteorenpuin en de ruimteschepen verschijnen op een willekeurige positie onderaan het scherm, en de informatie over hun posities wordt bewaard in de variabelen a, b, c, d en e, waarbij e het dichtst bij is. Door nu e te vergelijken met de positie van uw schip bepaalt het programma of er een crash dreigt.
In elke cyclus wordt er een SCROLL voor twee regels gegeven, dus er verschijnen slechts 11 meteoorbrokstukken per keer op het scherm. U zou iedere cyclus een SCROLL voor maar een regel kunnen geven, maar dan zou er veel te veel puin op het scherm komen en u zou twee keer zo veel variabelen in de gaten moeten houden.

Verbeteringen van het programma
Het programma is opzettelijk simpel gehouden om u de mogelijkheid te geven het te verbeteren.
Een eerste stap zou zijn om met de vrij programmeerbare karakters van de Spectrum prachtige vormen te ontwerpen voor het ruimteschip en de meteoorbrokstukken.
Verder zult u hebben gemerkt dat er twee plaatsen in het programma zijn waar geluid door de BEEP-instructie geproduceerd wordt. Het doel van deze instructies is niet alleen geluid te produceren! Veeg ze maar eens uit, dan zult u merken dat het programma veel te snel loopt.
U kunt dus het programma sneller of langzamer laten lopen door de lengte van de toon, of tonen, te veranderen.
Ook is genoeg tijd over in iedere cyclus om u de mogelijkheid te geven extra schepen die u aanvallen te programmeren, of de kans om op meteoren te schieten in te bouwen enz.

Programma

```
  10 REM asteroiden
 100 LET a$="*"
 110 LET n=0
 120 LET a=0
 130 LET b=0
 140 LET c=0
 150 LET d=0
 160 LET t=1
 170 LET y=12
 180 LET r=INT (27*RND)
 190 PRINT AT 21,r; INK (4*RND);a$
 200 POKE 23692,255
 210 PRINT AT 21,31;"##"
 220 BEEP .3,0
 230 PRINT AT 10,x-2;"#######"
 240 PRINT AT 11,x;"GRA 6 SHI 6 GRU"
 250 PRINT AT 21,31;"##"
 260 LET n=n+t
 270 IF n=100 THEN  LET a$="GRA SHI 6 SHI 7 GRU"
 280 IF n=104 THEN  LET t=2
 290 LET e=d
 300 LET d=c
 310 LET c=b
 320 LET b=a
 330 LET a=r
 340 PRINT AT 10,x-2;"#######"
 350 PRINT AT 11,x;"GRA 6 SHI 6 GRU"
 360 IF x>=e-2 AND x<=e+t THEN  GO TO 410
 370 BEEP .2,5
 380 IF INKEY$="5" THEN  LET x=x-t
 390 IF INKEY$="8" THEN  LET x=x+t
 400 GO TO 180
 410 PRINT AT 11,x-1;"CRASH"
 420 PRINT AT 0,0;"SCORE=";n
 430 BEEP 2,-12
 440 PRINT AT 21,0;"Druk#een#willekeurige#toets#in##voor#ee
n#nieuwe#poging"
 450 IF INKEY$="" THEN  GO TO 450
 460 RUN
```

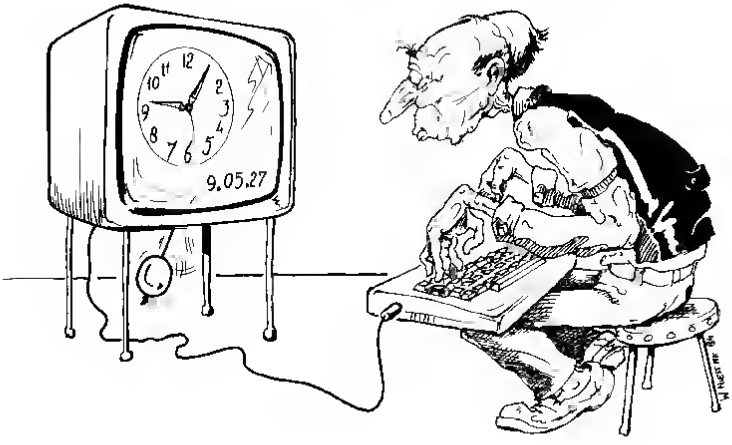
1.4 Spectrumklok
(Copyright Beam Software)

Beschrijving
Dit programma is een nabootsing van een 'real-time'-klok; zowel de gewone klok als de digitale klok worden bijgehouden op het scherm. U zult verbaasd zijn over de nauwkeurigheid van de Spectrumklok, en dat niet alleen: u kunt ook nog het alarm instellen en de Spectrumklok zal u daaraan herinneren door een toon te produceren.

Gebruik van het programma
Om het programma op te starten moet u RUN intikken gevolgd door ENTER. Dan moet u het uur, de minuut en seconde intikken om de klok gelijk te zetten. Denk er aan dat de uren op 24-uur basis bijgehouden worden, dus 1 uur 's middags is 13 uur. Het programma controleert dan of de invoer legaal is, u kunt bijvoorbeeld geen negatief getal invoeren of een getal groter dan 60 bij de minuten. Als u het alarm wilt instellen dan gaat dat op dezelfde manier.

Een mogelijkheid om het alarm doelmatig te gebruiken is om het programma als ochtendwekker te gebruiken: laat het programma 's nachts doorlopen nadat u de klok gelijk heeft gezet en het alarm ingesteld. De televisie kunt u atzetten om elektriciteit te besparen. De Spectrum zal het programma blijven doorlopen totdat de wektijd bereikt is, waarna de ingebouwde luidspreker een toon produceert om u te wekken.

Programma-opbouw
Het programma gebruikt de Spectrum beeldteller als interne klok. De beeldteller staat opgeslagen in drie bytes beginnende bij geheugenadres 23672 met de minst belangrijke



bytes vooraan. Elke 20 milliseconden wordt de teller met 1 verhoogd, met andere woorden, het totaal van wat er in de teller staat gedeeld door 50 geeft het aantal seconden dat verstreken is sinds met tellen begonnen is.

De opbouw van het programma is als volgt:

```
Initialisatie
  invoer van uur, minuut en seconde
  als alarm ingesteld wordt
    invoer alarmtijd
    stel alarmtijd in op scherm
  teken wijzerplaat
  bereken startwaarde van de beeldteller-bytes
  POKE-waarde in beeldteller bytes als toets ingedrukt wordt
  definieer functie: aantal seconden sinds start
  initialiseer starttijd
Hoofdlus
  U: bereken positie kleine wijzer
  M: bereken positie grote wijzer
  S: bereken positie secondewijzer
  teken secondewijzer
  teken grote wijzer
  teken kleine wijzer
  bereken de digitale klok en zet die op het scherm
    bereken uur, minuut en seconde, uitlezing
    aanpassen voor 12 uur 's middags-grens
    als alarmtijd overeenkomt met uitlezing
      flits klok en laat alarm klinken
  haal de beeldteller op totdat 1 seconde verstreken is
  teken nieuwe secondewijzer
  als minuut gelijk is gebleven, ga dan naar S
  teken nieuwe grote wijzer
  als uur gelijk is gebleven, ga dan naar M
  teken nieuwe kleine wijzer
  ga naar U
```

Speciale eigenschappen

DEF FN t() wordt gebruikt om veelvuldig de beeldteller te kunnen gebruiken. SIN- en COS-functies worden gebruikt om de wijzers te tekenen. De secondewijzer wordt elke seconde bijgesteld, de grote wijzer elke minuut en de kleine wijzer elke 12 minuten. Programma houdt rekening met voor- en namiddag-instelling. Programma controleert invoer op juistheid.

Speciale opmerkingen

De Spectrumklok is nauwkeurig tot op 10 seconden per dag, mits de computer niets anders doet dan dit programma te laten lopen, en geen invoer of uitvoer moet regelen of

geluid moet produceren, omdat de beeldteller niet bijgehouden wordt. Nadat het alarm is afgelopen moet de klok opnieuw gelijk worden gezet, dit betekent dat u het programma een RUN moet geven als u het weer wilt gebruiken.

Programma

```
  10 REM Klok
 100 GO SUB 360
 110 LET c=PI/30
 120 DEF FN t()=INT ((65536*PEEK 23674+256*PEEK 23673+PEEK
23672)/50): REM aantal seconden sinds start
 130 REM Nu starten we de klok
 140 LET t1=FN t()
 150 LET u1=INT (t1/720)
 160 LET u=u1*c
 170 LET ux=50*SIN u: LET uy=50*COS u
 180 LET m1=INT (t1/60)
 190 LET m=m1*c
 200 LET mx=60*SIN m: LET my=60*COS m
 210 LET a=t1*c: REM a is hoek seconden wijzer in radialen
 220 LET sx=72*SIN a: LET sy=72*COS a
 230 PLOT 131,91: DRAW sx,sy: REM teken seconden wijzer
 240 PLOT 131,91: DRAW mx,my
 250 PLOT 131,91: DRAW ux,uy
 260 GO SUB 740
 270 LET t=FN t()
 280 IF t<=t1 THEN  GO TO 270: REM wacht op tijd volgende w
ijzer
 290 LET t1=t
 300 PLOT 131,91: DRAW  OVER 1;sx,sy: REM  veeg seconden wi
jzer  uit
 310 IF INT (t1/60)<=m1 THEN  GO TO 210
 320 PLOT 131,91: DRAW mx,my: PLOT 131,91: DRAW  OVER 1;mx,
my
 330 IF INT (t1/720)<=u1 THEN  GO TO 180
 340 PLOT 131,91: DRAW ux,uy: PLOT 131,91: DRAW  OVER 1;ux,
uy
 350 GO TO 150
 360 BORDER 7: PAPER 7: INK 0: OVER 0: FLASH 0: CLS : PRINT
 "Welk#uur#svp?": PRINT "(0#-#23)"
 370 INPUT U: IF U<0 OR U>23 THEN  GO TO 370
 380 PRINT  FLASH 1; INK 2;U
 390 LET D=0: IF U>=12 THEN  LET D=1: LET U=U-12*(U>12)
 400 PRINT 'Welke#minuut?": PRINT "(0#-#59)"
 410 INPUT M: IF M<0 OR M>59 THEN  GO TO 410
 420 PRINT  FLASH 1; INK 1;M
 430 PRINT 'Welke#seconde?": PRINT "(0#-#59)"
 440 INPUT S: IF S<0 OR S>59 THEN  GO TO 440
 450 PRINT  FLASH 1; INK 3;S
 460 PAUSE 50: CLS
 470 PRINT 'Wil#je#het#ALARM#zetten?": PRINT "(j#of#n)"
 480 INPUT a$: IF a$<>"j" AND a$<>"n" THEN  GO TO 480
 490 IF a$="n" THEN  LET a1=0: BORDER 0: PAPER 0: INK 7: CL
S : GO TO 630
```

```
 500 LET al=1: PRINT : PRINT "Uur?#";
 510 INPUT au: IF au<0 OR au>24 THEN  GO TO 510
 520 PRINT au: LET ad=0: IF au>11 THEN  LET ad=1: LET au=au
-(au<>12)*12
 530 PRINT : PRINT "Minuut?#";
 540 INPUT am: IF am<0 OR am>59 THEN  GO TO 540
 550 PRINT am
 560 PRINT : PRINT "Seconde?#";
 570 INPUT as: IF as<0 OR as>59 THEN  GO TO 570
 580 PRINT as: PAUSE 50
 590 REM Teken wijzerplaat
 600 PAPER 0: BORDER 0: INK 7: CLS : PRINT AT 20,0;"ALARM":
 PRINT au;":";am;":";as;"#";
 610 IF ad=0 THEN  PRINT "VM": GO TO 630
 620 IF ad=1 THEN  PRINT "NM"
 630 FOR n=1 TO 12
 640 PRINT AT 10-10*COS (n/6*PI),16+10*SIN (n/6*PI);n
 650 NEXT n
 660 LET dh=0: LET dm=M: LET ds=S
 670 LET T=(U*3600+M*60+S)*50
 680 LET b1=T-INT (T/256)*256
 690 LET b2=INT (T/256)-INT (T/256^2)*256
 700 LET b3=INT (T/256^2)-INT (T/256^3)*256
 710 IF INKEY$="" THEN  GO TO 710
 720 POKE 23674,b3: POKE 23673,b2: POKE 23672,b1
 730 RETURN
 740 LET ds=t1-INT (t1/60)*60: LET dm=INT (t1/60)-INT (t1/3
600)*60: LET du=INT (t1/3600)
 750 IF ds=0 THEN  PRINT AT 0,0;"########"
 760 IF du>=13 THEN  LET du=du-12
 770 IF du<>12 OR ds<>0 OR dm<>0 THEN  GO TO 810
 780 IF D=0 THEN  LET D=1: LET du=12: GO TO 810
 790 IF D=1 THEN  LET D=0: LET du=0
 800 LET du=du-(du=12 AND D=0)*12
 810 PRINT AT 0,0;du;":";dm;":"; FLASH 1;ds
 820 PRINT AT 0,8;"#";
 830 IF D=0 THEN  PRINT "VM"
 840 IF D=1 THEN  PRINT "NM"
 850 IF al=0 THEN  RETURN
 860 IF au<>du OR am<>dm OR as<>ds OR ad<>D THEN  RETURN
 870 PRINT AT 0,0;du;":";dm;":";ds: FOR s=22528 TO 23231: P
OKE s,128+PEEK s: NEXT s
 880 BEEP 0.5,27: BEEP 0.5,20: IF INKEY$="" THEN  GO TO 880

 890 RETURN
```

1.5 3-D Doolhof

(Copyright Beam Software)

U zit vast in een doolhof. U moet ontsnappen! De druk wordt steeds groter. U loopt rechtdoor, plotseling is er links een deuropening, snel! Oh nee, het loopt alweer dood! Zult u hier ooit levend uitkomen?
Deze verbazingwekkende 3-dimensionale nabootsing plaatst u midden in het doolhof met de taak om de uitgang te vinden. U mag dankbaar zijn dat er geen monsters zijn!! Nog niet!!

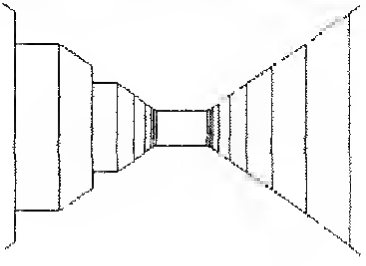
U mag er zo lang over doen als u wilt, maar de tijd die u nodig heeft wordt bijgehouden. Op elke plaats in het doolhof krijgt u een zeer realistisch beeld te zien. U kunt links- of rechtsaf (resp. 'o' en de 'p'-toets), rechtdoor (de SPACE-toets) of omdraaien (de 'r'-toets).
Als u werkelijk verdwaalt kan het een goed idee zijn om een tekening te maken.

Programma-opbouw

Dit programma gebruikt de PLOT- en DRAW-functies van de Spectrum.
Op elk punt in de gang bestaat er de mogelijkheid dat de deur dicht of open is. Dit programma gebruikt verschillende subroutines om de deur te tekenen, afhankelijk van de afstand tot de kijker en of de deur open of dicht is.
Dit verklaart de verschillende subroutines van regel 6000 tot 8508.
Een afzonderlijke data-tabel (regel 9100 - 9208) zet het specifieke doolhof op waarin u zich op dat moment bevindt.

Uw eigen doolhof opzetten

Het doolhof dat hier wordt gebruikt, wordt op een 9 bij 9 raster getekend, zo iets als een schaakbord, en elke positie in het doolhof wordt weergegeven door een hokje op het bord.
Het is duidelijk dat elk hokje in elk van de vier richtingen een muur kan hebben. Dit programma gebruikt twee arrays om de muren bij te houden, een array voor de muren in de verticale richting op het bord, en een array voor de muren in horizontale richting.
In totaal moeten 18 lijnen gedefinieerd worden, 9 verticaal en 9 horizontaal. In elke lijn zet u een 0 als er geen muur staat om u de weg te blokkeren, en een 1 als er wel een muur is.
Het geheel is altijd omsloten door muren, dus deze hoeven niet gedefinieerd te worden.
Deze informatie is opgeslagen in DATA-statements, regel 9100-9108 en 9200-9208.
Als u bijvoorbeeld vanaf het hokje linksonder omhoog kijkt, dan ziet u een muur. Als u door



die muur heen zou kunnen breken, dan zou er weer een nieuwe staan, en dan weer een Uiteindelijk zou u een gang tegen komen die zich over 4 hokjes uitstrekt tot aan de laatste muur. Al deze informatie staat in regel 9100.

De uitgang wordt in regel 156 vast gelegd. Deze regel stelt dat als u zich in de negende baan bevindt, u omgedraaid bent, en minder dan 5 hokjes verwijderd bent van het einde, u het bord met UITGANG te zien krijgt. De voorwaarde van 5 hokjes van het einde is nodig omdat er een muur staat in dit doolhof tussen hokje 6 en 5 in baan 9.

U kunt het programma gemakkelijk aanpassen aan een eigen doolhof, maar vergeet niet de voorwaarden voor UITGANG te veranderen.

Programma

```
  10 REM doolhof
 100 DIM v(10,10): DIM h(10,10)
 110 GO SUB 9000
 120 LET x=1: LET y=1: LET dx=1: LET dy=0
 125 LET ti=PEEK 23672+256*PEEK 23673+4096*PEEK 23674
 130 LET l=9: LET lx=x+(dx=-1): LET ly=y+(dy=-1)
 132 LET l=l-1: LET lx=lx+dx: LET ly=ly+dy
 134 IF dx<>0 AND v(lx,ly)=0 THEN  GO TO 132
 136 IF dy<>0 AND h(lx,ly)=0 THEN  GO TO 132
 140 CLS : GO SUB 6000+l
 145 LET lx=lx-(dx=-1): LET ly=ly-(dy=-1)
 150 FOR i=1 TO 8
 155 LET lx=lx-dx: LET ly=ly-dy: GO SUB 1000: NEXT i
 156 IF dx=-1 AND y=9 AND x<5 THEN  PRINT AT 10,14;"EXIT SHIFT
 2 UITGANG EXTEND MODE 0"
 158 LET a$=INKEY$: IF a$="" THEN  GO TO 158
 160 IF a$<>"#" THEN  GO TO 190
 170 IF (dx=1 AND v(x+1,y)=0) OR (dx=-1 AND v(x,y)=0) OR (d
y=1 AND h(x,y+1)=0) OR (dy=-1 AND h(x,y)=0) THEN  LET x=x+d
x: LET y=y+dy
 190 IF x=1 AND y=9 THEN  GO TO 9990
 200 IF a$="r" THEN  LET dx=-dx: LET dy=-dy: GO TO 130
 210 IF a$="o" THEN  GO TO 300
 215 IF a$<>"p" THEN  GO TO 130
 220 IF ABS dx=1 THEN  LET dy=-dx: LET dx=0: GO TO 130
 230 LET dx=dy: LET dy=0: GO TO 130
 300 IF ABS dx=1 THEN  LET dy=dx: LET dx=0: GO TO 130
 310 LET dx=-dy: LET dy=0: GO TO 130
1000 IF dx<>1 THEN  GO TO 1100
1010 IF h(lx,ly+1)=0 THEN  GO SUB 7500+i: GO TO 1050
1020 GO SUB 7000+i
1050 IF h(lx,ly)=0 THEN  GO TO 8500+i
1060 GO TO 8000+i
1100 IF dx<>-1 THEN  GO TO 1200
1110 IF h(lx,ly)=0 THEN  GO SUB 7500+i: GO TO 1150
1120 GO SUB 7000+i
1150 IF h(lx,ly+1)=0 THEN  GO TO 8500+i
1160 GO TO 8000+i
```

```
1200 IF dy<>-1 THEN  GO TO 1300
1210 IF v(lx+1,ly)=0 THEN  GO SUB 7500+i: GO TO 1250
1220 GO SUB 7000+i
1250 IF v(lx,ly)=0 THEN  GO TO 8500+i
1260 GO TO 8000+i
1300 IF v(lx,ly)=0 THEN  GO SUB 7500+i: GO TO 1350
1310 GO SUB 7000+i
1350 IF v(lx+1,ly)=0 THEN  GO TO 8500+i
1360 GO TO 8000+i
6000 PLOT 110,76: DRAW 0,24: DRAW 36,0: DRAW 0,-24: DRAW -3
6,0: RETURN
6001 PLOT 108,75: DRAW 0,26: DRAW 40,0: DRAW 0,-26: DRAW -4
0,0: RETURN
6002 PLOT 106,74: DRAW 0,28: DRAW 44,0: DRAW 0,-28: DRAW -4
4,0: RETURN
6003 PLOT 102,71: DRAW 0,34: DRAW 52,0: DRAW 0,-34: DRAW -5
2,0: RETURN
6004 PLOT 94,65: DRAW 0,46: DRAW 68,0: DRAW 0,-46: DRAW -68
,0: RETURN
6005 PLOT 82,57: DRAW 0,62: DRAW 92,0: DRAW 0,-62: DRAW -92
,0: RETURN
6006 PLOT 64,45: DRAW 0,86: DRAW 128,0: DRAW 0,-86: DRAW -1
28,0: RETURN
6007 PLOT 40,29: DRAW 0,118: DRAW 176,0: DRAW 0,-118: DRAW
-176,0: RETURN
6008 PLOT 8,7: DRAW 0,162: DRAW 240,0: DRAW 0,-162: DRAW -2
40,0: RETURN
7000 PLOT 108,75: DRAW 2,1: DRAW 0,24: DRAW -2,1: RETURN
7001 PLOT 106,74: DRAW 2,1: DRAW 0,26: DRAW -2,1: RETURN
7002 PLOT 102,71: DRAW 4,3: DRAW 0,28: DRAW -4,3: RETURN
7003 PLOT 94,65: DRAW 8,6: DRAW 0,34: DRAW -8,6: RETURN
7004 PLOT 82,57: DRAW 12,8: DRAW 0,46: DRAW -12,8: RETURN
7005 PLOT 64,45: DRAW 18,12: DRAW 0,62: DRAW -18,12: RETURN

7006 PLOT 40,29: DRAW 24,16: DRAW 0,86: DRAW -24,16: RETURN

7007 PLOT 8,7: DRAW 32,22: DRAW 0,118: DRAW -32,22: RETURN

7008 PLOT 0,1: DRAW 8,6: DRAW 0,162: DRAW -8,6: RETURN
7500 PLOT 108,76: DRAW 2,0: DRAW 0,24: DRAW -2,0: RETURN
7501 PLOT 106,75: DRAW 2,0: DRAW 0,26: DRAW -2,0: RETURN
7502 PLOT 102,74: DRAW 4,0: DRAW 0,28: DRAW -4,0: RETURN
7503 PLOT 94,71: DRAW 8,0: DRAW 0,34: DRAW -8,0: RETURN
7504 PLOT 82,65: DRAW 12,0: DRAW 0,46: DRAW -12,0: RETURN
7505 PLOT 64,57: DRAW 18,0: DRAW 0,62: DRAW -18,0: RETURN
7506 PLOT 40,45: DRAW 24,0: DRAW 0,86: DRAW -24,0: RETURN
7507 PLOT 8,29: DRAW 32,0: DRAW 0,118: DRAW -32,0: RETURN
7508 PLOT 0,7: DRAW 8,0: DRAW 0,162: DRAW -8,0: RETURN
8000 PLOT 148,75: DRAW -2,1: DRAW 0,24: DRAW 2,1: RETURN
8001 PLOT 150,74: DRAW -2,1: DRAW 0,26: DRAW 2,1: RETURN
8002 PLOT 154,71: DRAW -4,3: DRAW 0,28: DRAW 4,3: RETURN
8003 PLOT 162,65: DRAW -8,6: DRAW 0,34: DRAW 8,6: RETURN
8004 PLOT 174,57: DRAW -12,8: DRAW 0,46: DRAW 12,8: RETURN

8005 PLOT 192,45: DRAW -18,12: DRAW 0,62: DRAW 18,12: RETUR
N
```

```
8006 PLOT 216,29: DRAW -24,16: DRAW 0,86: DRAW 24,16: RETUR
N
8007 PLOT 248,7: DRAW -32,22: DRAW 0,118: DRAW 32,22: RETUR
N
8008 PLOT 255,1: DRAW -8,6: DRAW 0,162: DRAW 8,6: RETURN
8500 PLOT 148,76: DRAW -2,0: DRAW 0,24: DRAW 2,0: RETURN
8501 PLOT 150,75: DRAW -2,0: DRAW 0,26: DRAW 2,0: RETURN
8502 PLOT 154,74: DRAW -4,0: DRAW 0,28: DRAW 4,0: RETURN
8503 PLOT 162,71: DRAW -8,0: DRAW 0,34: DRAW 8,0: RETURN
8504 PLOT 174,65: DRAW -12,0: DRAW 0,46: DRAW 12,0: RETURN

8505 PLOT 192,57: DRAW -18,0: DRAW 0,62: DRAW 18,0: RETURN

8506 PLOT 216,45: DRAW -24,0: DRAW 0,86: DRAW 24,0: RETURN

8507 PLOT 248,29: DRAW -32,0: DRAW 0,118: DRAW 32,0: RETURN

8508 PLOT 255,7: DRAW -7,0: DRAW 0,162: DRAW 7,0: RETURN
9000 FOR i=1 TO 9: LET h(i,1)=1: LET h(i,10)=1: FOR j=2 TO
9: READ a: LET h(i,j)=a: NEXT j: NEXT i
9010 FOR i=1 TO 9: LET v(1,i)=1: LET v(10,i)=1: FOR j=2 TO
9: READ a: LET v(j,i)=a: NEXT j: NEXT i
9020 RETURN
9100 DATA 1,1,1,0,0,0,0,1
9101 DATA 0,0,1,0,0,1,0,1
9102 DATA 1,1,0,0,0,0,1,1
9103 DATA 0,0,1,1,0,0,1,0
9104 DATA 1,1,1,0,0,1,1,0
9105 DATA 1,0,0,0,0,0,0,1
9106 DATA 1,1,1,0,1,0,0,1
9107 DATA 1,1,1,0,0,1,1,0
9108 DATA 1,0,0,0,0,0,0,0
9200 DATA 0,0,0,0,0,0,0,0
9201 DATA 1,1,1,1,0,1,0,0
9202 DATA 0,0,0,0,0,0,0,1
9203 DATA 0,1,0,0,1,0,0,0
9204 DATA 1,1,0,1,1,1,1,1
9205 DATA 1,1,1,1,1,1,0,1
9206 DATA 0,0,0,0,0,0,1,0
9207 DATA 1,1,0,0,1,1,0,0
9208 DATA 0,0,0,1,0,0,0,1
9990 LET te=PEEK 23672+256*PEEK 23673+4096*PEEK 23674
9991 LET t=te-ti
9992 PRINT AT 0,4;"GOED#GEDAAN!#Je#hebt#het###gehaald"
9993 PRINT AT 1,8;"in#";t/50;"#seconden"
9994 STOP
```

2. Onderwijskundige programma's

2.1 Meetkundige test

(Copyright Beam Software)

Toevallig beslaan de testvragen uit dit programma het terrein van de meetkunde, maar de opzet van het programma is algemener. Dit programma kan in elke situatie gebruikt worden waar een serie meerkeuzevragen nodig is.

Een interessant gegeven van de Spectrum is dat bij een wiskundige test zowel een getal als een formule als antwoord geaccepteerd wordt. Bijvoorbeeld op de vraag 'Wat is de oppervlakte van een cirkel met straal 2?' zou zowel het antwoord '4 x PI' als '12.75' goedgekeurd worden. (Het laatste antwoord wordt geaccepteerd, hoewel het niet helemaal correct is, omdat het programma met een afronding van .5 genoegen neemt.)

Programma-opbouw

Een korte controlelus in regel 100-260 kiest de vraag, vergelijkt het antwoord met wat het zou moeten zijn en zet van toepassing zijnde mededeling op het scherm.
Elke vraag staat in een subroutine die niet alleen de vraag op het scherm zet, maar ook het antwoord specificeert dat goedgekeurd wordt.
Het moet duidelijk zijn dat het aantal en soort vragen alleen maar gelimiteerd wordt door uw voorstellingsvermogen.

Programma

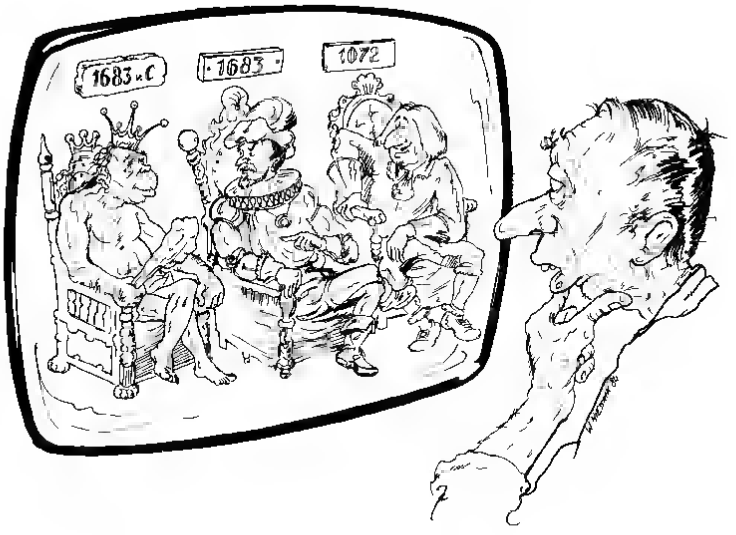
```
  10 REM meetkunde
 100 CLS
 110 LET q=1+INT (4*RND)
 120 LET m=2+INT (20*RND)
 130 GO SUB q*1000
 200 INPUT r: PRINT r
 210 IF ABS (r-a)>.5 THEN  GO TO 240
 220 PRINT "JA,#het#antwoord#is#";a
 230 GO TO 250
 240 PRINT "SORRY,#het#antwoord#was#";a
 250 INPUT "<ENTER>#als#je#klaar#bent#';y$
 260 RUN
1000 PRINT "Wat#is#de#omtrek#van#een#cirkel#met#straal#';m
1010 LET a=2*PI*m
1020 RETURN
2000 PRINT "Wat#is#de#oppervlakte#van#een###cirkel#met#stra
al#";m
2010 LET a=PI*m*m
2020 RETURN
3000 PRINT "Wat#is#de#buitenoppervlakte#van#een#bol#met#str
aal#';m
3010 LET a=(4/3)*PI*m*m
3020 RETURN
4000 PRINT "Wat#is#de#inhoud#van#een#bol#met straal#';m
4010 LET a=PI*m*m*m/3
4020 RETURN
```

2.2 Het huis van Oranje

Dit programma gebruikt de meerkeuzetest-techniek om uw kennis van de Nederlandse geschiedenis op te frissen.

De computer kiest een jaartal tussen 1544 en 1980 en vraagt dan wie er in dat jaar het Huis van Oranje regeerde. U kunt kiezen tussen drie namen. Als er in het desbetreffende jaar meer dan één persoon regeerde dan geldt als juist antwoord de eerste regent in dat jaar. Tik uw keuze in, alleen de computer weet zeker of u het fout of goed heeft!

Na 25 vragen beantwoord te hebben krijgt u het resultaat in procenten op het scherm te zien. Als u bij elke vraag het antwoord gegokt zou hebben, dan heeft u een kans van 1 op 3 gehad. U moet dus aardig uw best doen wilt u minder dan 30% scoren.



Programma-opbouw

De opbouw van het programma is betrekkelijk eenvoudig, maar de manier waarop de gegevens opgeslagen zijn is de moeite van het bestuderen waard.

Daar er voor elk jaartal maar één goed antwoord mogelijk is, door de manier waarop de vragen geformuleerd zijn, kunt u gebruik maken van de mogelijkheid om op de Spectrum een subroutine aan te roepen met een variabele in plaats van een regelnummer.

Het komt u ook goed uit dat bij het aanroepen van een subroutine op een niet-bestaand regelnummer de Spectrum het volgende regelnummer zal produceren enz.

Bekijk de uitdraai van het programma - u ziet dat er geen regelnummers voorkomen tussen regel 1586 en 1625. Als de opdracht GOSUB 1600 wordt gegeven dan gaat het programma naar regel 1600, deze bestaat niet, het programma probeert regel 1601 enz. net

zo lang totdat hij regel 1625 gevonden heeft. Een GOSUB met een getal tussen 1587 en 1625 levert dus altijd het volgende resultaat op

a$ = Maurits

Dat is al heel wat, want Maurits regeerde tussen 1585 en 1625 en voor deze test was elk jaar tussen 1586 en 1625 correct

Wat zou er nu met een GOSUB 1586 gebeuren?
Op regel 1586 staat alleen maar een RETURN-instructie. De aangeroepen subroutine komt dus terug zonder informatie over a$.
Dit probleem wordt opgelost in regel 530 door de subroutine naar een volgende regel te sturen als a$ leeg is. In dit geval is dat regel 1587 en krijgt u toch nog Maurits als antwoord.
Deze programmastructuur zou voor elke andere test, waarin elkaar niet-overlappende waarden voorkomen, gebruikt kunnen worden

Het Huis van Oranje:

1544-1585	Willem van Oranje	1751 1795	Willem V
1585-1625	Maurits	1795-1815	geen Oranje regent
1625-1647	Frederik Hendrik	1815-1840	Willem I
1647-1650	Willem II	1840-1849	Willem II (2)
1650-1672	geen Oranje regent	1849-1890	Willem III (2)
1672-1702	Willem III	1890-1948	Wilhelmina
1702-1711	Johan Willem Friso	1948-1980	Juliana
1711-1751	Willem IV		

Programma

```
 10 REM oranje
100 DIM p(3)
110 RANDOMIZE
120 LET s=0
130 FOR i=1 TO 25
140 GO SUB 500
150 LET c$=a$
160 GO SUB 500
170 LET b$=a$
180 IF b$=c$ THEN  GO TO 160
190 GO SUB 500
200 IF a$=b$ OR a$=c$ THEN  GO TO 190
210 PRINT i;'.#Wie#regeerde#in#';n;'#?"
220 LET p(1)=INT (3*RND)+1
230 LET p(2)=INT (3*RND)+1
240 IF p(2)=p(1) THEN  GO TO 230
250 LET p(3)=6-p(1)-p(2)
260 FOR t=1 TO 3
```

```
 270 PRINT AT 5+t,5;t;".#";
 280 IF t=p(1) THEN  PRINT a$
 290 IF t=p(2) THEN  PRINT b$
 300 IF t=p(3) THEN  PRINT c$
 310 NEXT t
 320 PRINT AT 12,0;
 330 LET x$= INKEY$: IF x$<"1" OR x$>"3" THEN  GO TO 330
 340 IF VAL x$=p(1) THEN  PRINT "PRIMA": LET s=s+1: BEEP .5
,2: BEEP .2,5: BEEP .2,7: GO TO 370
 350 PRINT "Nee#-#het#was#";a$
 360 FOR j=1 TO 8: BEEP RND/4,10*RND-30: NEXT j: BEEP 1,-25

 370 PRINT  FLASH 1;AT 20,0;"######druk#een#toets#in#######
#"
 380 LET x$=INKEY$
 390 IF x$="" THEN  GO TO 380
 400 CLS
 410 NEXT i
 420 PRINT "JOUW#SCORE#IS#";4*s;"#PROCENT"
 430 STOP
 500 LET n=INT (426*RND)+1544
 510 LET a$="#"
 520 GO SUB n
 530 IF a$="#" THEN  GO SUB n+1
 540 RETURN
1585 LET a$="Willem#van#Oranje"
1586 RETURN
1625 LET a$="Maurits"
1626 RETURN
1647 LET a$="Frederik#Hendrik"
1648 RETURN
1650 LET a$="Willem#II"
1651 RETURN
1672 LET a$="geen#Oranje-regent"
1673 RETURN
1702 LET a$="Willem#III"
1703 RETURN
1711 LET a$="Johan#Willem#Friso"
1712 RETURN
1751 LET a$="Willem#IV"
1752 RETURN
1795 LET a$="Willem#V"
1796 RETURN
1815 LET a$="geen#Oranje-regent"
1816 RETURN
1840 LET a$="Willem#I"
1841 RETURN
1849 LET a$="Willem#II#(2)"
1850 RETURN
1890 LET a$="Willem#III#(2)"
1891 RETURN
1948 LET a$="Wilhelmina"
1949 RETURN
1980 LET a$="Juliana"
1981 RETURN
```

3. Gokspelletjes

3.1 Blackjack
(Copyright Beam Software)

Dit is het bekende casinospel bij de Club ZX-Spectrum. Welkom! Als een speciale groet aan al onze gasten krijgt u van ons een tegoedbon voor $ 100, alleen geldig aan onze blackjack-tafels.
Komt u rustig meespelen. De huisregels zijn als volgt:

- voordat de kaarten worden verdeeld moet u hebben besloten hoeveel u wilt inzetten. Denk er wel aan dat de club ZX Spectrum uw krediet niet verlengt.
- de bank geeft zichzelf een kaart open op tafel, dan krijgt u er ook een. U mag daarna net zoveel kaarten bijvragen als u zelf wilt.
- als u de 21 punten niet overschreden heeft, dan geeft de bank zichzelf kaarten bij. De bank pakt altijd een nieuwe kaart bij 16 punten of minder, bij 17 of meer past de bank.

Als u wint dan verdubbelt de bank uw inzet, als u gelijk speelt dan wint de bank. Als u met 5 kaarten onder de 21 punten blijft dan heeft u gewonnen, mits de bank geen blackjack heeft (21 punten met 2 kaarten). Als u zelf blackjack heeft dan krijgt u uw inzet verdubbeld terug, mits natuurlijk de bank zelf weer geen blackjack heeft.
Zet u in dames en heren!

Programma-opbouw
De opbouw is als volgt:

```
Nieuwe inzet
  invoer inzet nieuwe speler
  als er geen geld meer over is, dan stop
  geef eerste kaart aan bank en speler

Voor speler en bank
  als beurt aan speler is, vraag dan of hij nieuwe kaart wil
  als niet, ga dan naar bank
  geef kaart en zet deze op scherm
  bereken de waarde van beide kaarten en zet deze op scherm
  als dit meer is dan 21, ga dan naar Uitbetaling
  als bank meer dan 16 heeft, ga dan naar Uitbetaling

Uitbetaling
  als speler meer dan 21 of bank meer dan of gelijk aan speler, dan uitbetaling = 0
  als Blackjack dan verdubbel uitbetaling
  pot = pot + uitbetaling
  ga weer naar Nieuwe inzet
```

De opbouw van het programma is redelijk eenvoudig, u zou dus weinig problemen moeten hebben om het programma aan te passen als u de huisregels wilt wijzigen.

Grafisch werk
De subroutine op regel 8000 definieert het ontwerp van de vier kaartsoorten die in 1 karakter passen. Deze subroutine wordt natuurlijk maar een keer aangeroepen, namelijk aan het begin van het programma.
Iedere kaart wordt door de subroutine op regel 8500 op het scherm gezet. Eerst de randen van de kaart in lichtblauw en dan het ene karakter dat de soort aangeeft in de benodigde kleur (zwart of rood).

Dan heeft u het probleem hoe u de kaarten zelf op het scherm moet krijgen. Zoals u weet heeft u slechts de beschikking over 21 vrij programmeerbare karakters. Daarvan heeft u er al 4 gebruikt om de soort aan te geven, zodat u er slechts 17 over heeft.
Er komen in het spel 13 verschillende kaarten voor, maar als u die in slechts één programmeerbaar karakter zou stoppen, dan zouden ze net zo groot worden als een normaal karakter. U lost dit probleem hier op door de vaste vorm van elk karakter op te zoeken in de ROM waar ze opgeslagen zijn. Vervolgens vergroot u deze vorm twee keer (elke vorm wordt dan twee karakters breed en twee hoog) en slaat die informatie op in de programmeerbare karakterset als ABCD. Omdat er geen standaard karakter '10' is, ontwerpt u hiervoor een nieuw karakter in subroutine 8000.
Deze methode is een beetje langzaam, maar in dit programma wel acceptabel omdat het de spanning verhoogt van het wachten op welke kaart u gedeeld krijgt.

Zoals u misschien wel weet wordt elk karakter op het scherm gezet als een groep van 8 bytes en staan de definities voor de vaste karakters opgeslagen in de ROM te beginnen op adres 15616. Het startadres voor ieder karakter kan van de bijbehorende code afgeleid worden aan de hand van de formule 15360 + 8 * karaktercode (zie regel 9020).
Het programma moet dan aan het werk om uit te zoeken hoe deze vorm twee keer dient te worden vergroot, om die informatie daarna in de programmeerbare karakterset op te slaan.

Opmerkingen
In regel 270 zult u een vreemde notering tegen komen,

```
LET a = p(i) = 11
```

In dit programma wordt de variabele 'a' gebruikt om het aantal azen in een hand bij te houden. Wat deze regel doet is:

```
LET a = 0
IF p(i) = 11 THEN LET a = 1
```

waarbij p(i) het aantal kaarten van speler i is.
De uitdrukking zou als hierboven geschreven kunnen worden, of als

```
LET a = ( p(i) = 11)
```

omdat u weet dat de ZX-Spectrum een uitdrukking als 0 evalueert als deze niet waar is en als 1 als deze waar is. Het blijkt nu dat u de haakjes weg kunt laten zonder kans op fouten en zo krijgt u de vreemde doch zeer compacte uitdrukking van regel 270.

Een andere uitdrukking die u wat eigenaardig zou kunnen vinden komen we tegen in regel 530 waar de waarde van p(i) berekend wordt. We hebben eerst een kaartnummer gekozen tussen de 2 en 14, dit getal legt u vast in variabele 'c', en u wilt nu de waarde van c erbij optellen als c kleiner is dan of gelijk is aan 10 (een 'lage' kaart). 10 erbij optellen als c een 'plaatje' aangeeft (c = 12, 13 of 14) en 1 erbij optellen als c een aas aangeeft (c = 11).
Dit kunnen we doen door te zeggen

IF c = < 10 THEN ..
IF c > 10 THEN ..
IF c = 11 THEN ..

maar die ene regel 270 doet het bijna allemaal in een keer. Die vertelt het programma de waarde van c erbij op te tellen als c kleiner is dan 12, en er 10 bij op tellen als c groter is dan 11. Simpel nietwaar!

Regels 540-560 zorgen ervoor dat de aas als 1 of als 11 geteld wordt en vereenvoudigen het programma daardoor aanzienlijk.

Programma

```
  10 REM blackjack
 100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
 110 LET b$="N2345678910AJQK"
 120 LET a=0: LET m=100
 130 GO SUB 8000
 140 DIM p(4)
 150 IF m=0 THEN  PRINT AT 10,1;"De#pot#bevat#$0";AT 19,0;"
Sorry,#uw#krediet#is#op.#We#zienu#graag#terug#als#u#meer#ge
ld###heeft.": STOP
 160 PRINT AT 10,1;"De#pot#bevat#$";m: PRINT "Hoeveel#zet#u
#in?"
 170 INPUT b
 180 CLS
 190 IF b>m THEN  GO TO 160
 200 LET m=m-b
 210 PRINT  INK 0;AT 0,0;"SPELER";AT 11,0;"SPECTRUM"
 220 PRINT AT 9,1;"Totaal=";AT 20,1;"Totaal="
 230 FOR i=2 TO 1 STEP -1
 240 GO SUB 500
 250 NEXT i
 260 FOR i=1 TO 2
 270 LET a=p(i)+11
```

```
 280 IF i=1 THEN  INPUT  FLASH 1; PAPER 2; INK 7;"Wilt#u#no
g#een#kaart?#";a$: LET a$=a$+"#": IF a$(1)="n" THEN  GO TO
340
 290 GO SUB 500
 300 IF p(i)>21 THEN  LET i=2: GO TO 340
 310 IF p(i+2)=5 THEN  GO TO 340
 320 IF i=2 AND p(i)>16 THEN  GO TO 340
 330 GO TO 280
 340 NEXT i
 350 IF p(1)<=21 AND p(3)=5 AND (p(2)<>21 OR p(4)<>2) THEN
 GO TO 380
 360 IF p(1)=21 AND p(3)=2 AND (p(2)<>21 OR p(4)<>2)) THEN
LET b=b*1.5: PRINT  FLASH 1;AT 5,18;"BLACKJACK": GO TO 380
 370 IF (p(2)<=21 AND p(2)>=p(1)) OR p(1)>21 THEN  LET b=0
 380 LET m=m+2*b
 390 GO TO 130
 500 LET p(i+2)=p(i+2)+1
 510 LET c=INT (13*RND)+2
 520 IF c=11 THEN  LET a=a+1
 530 LET p(i)=p(i)+c*(c<12)+10*(c>11)
 540 IF p(i)<22 OR a=0 THEN  GO TO 570
 550 LET a=a-1
 560 LET p(i)=p(i)-10
 570 GO SUB 8500
 580 GO SUB 9000
 590 PRINT  INK 0;AT 11*i-2,9;p(i)
 600 RETURN
8000 DATA 0,78,209,81,81,81,78,0
8010 DATA 102,255,255,255,126,60,24,24
8020 DATA 24,60,126,255,255,126,60,24
8030 DATA 24,60,126,255,255,219,24,60
8040 DATA 60,60,219,231,231,219,24,24
8050 RESTORE 8000
8060 FOR k=USR "e" TO USR "e"+39: READ x: POKE k,x: NEXT k
8070 RETURN
8500 LET p=11*i-7: LET q=6*p(i+2)-5
8510 INK 5: PRINT AT p-3,q;"GRA 5 3 3 3 3 SHI 5 GRU": FOR k
=1 TO 5: PRINT AT p-3+k,q;"GRA 5 #### SHI 5 GRU": NEXT k: P
RINT AT p+3,q;"GRA 1 3 3 3 3 2 GRU": INK 0
8520 LET suit=1+INT (4*RND): INK 2*(suit<3): PRINT AT p-2,q
+3;CHR$ (148+suit)
8530 RETURN
9000 LET x=USR "a"
9010 FOR k=0 TO 31: POKE x+k,0: NEXT k
9020 LET s=8*CODE b$(c)+15360: IF b$(c)="T" THEN  LET s=USR
 "e"
9030 FOR j=0 TO 7: LET v=PEEK (s+j): LET c=0: LET d=256
9040 FOR k=8 TO 1 STEP -1: LET d=d/2: IF v>=d THEN  LET c=c
+7*d*d: LET v=v-d
9050 NEXT k
9060 LET h=INT (c/256): LET l=c-256*h
9070 POKE x,h: POKE x+1,h: POKE x+8,l: POKE x+9,l
9080 LET x=x+2+8*(j=3)
9090 PRINT AT p,q+2;"GRA A B GRU";AT p+1,q+2;"GRA C D GRU"
9100 NEXT j
9110 RETURN
```

3.2 Fruitmachine

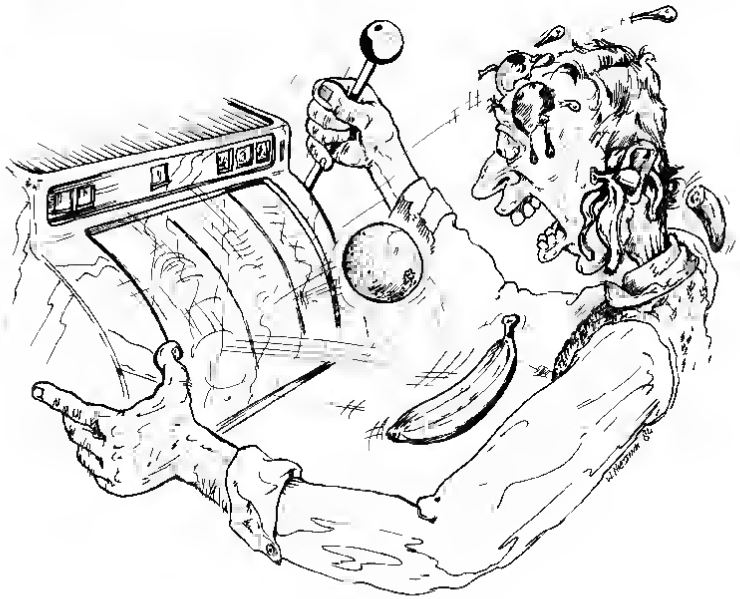
(Copyright Beam Software)

Dit programma bootst een fruitmachine na zoals u die in een cafe of een gokhal kunt vinden
Het grote voordeel van thuisgokken is natuurlijk dat u de 100 guiden die u bij zich had niet kwijt bent als u verliest. Maar het is natuurlijk ook een nadeel dat u de winst niet kunt opstrijken.

Het programma Fruitmachine verrijkt waarschijnlijk uw inzicht in de toepassingsmogelijkheden van vrij programmeerbare karakters. In dit programma zult u 60 karakters definieren voor de 16K versie. Maar op een 48K-machine zou u nog veel meer grafische karakters kunnen programmeren zonder al te veel extra moeite.
Doet u dus rustig mee dames en heren en probeer uw geluk!
U zult op zijn minst onder de indruk zijn van de grafische mogelijkheden van dit programma.
Het is duidelijk dat u wint als alle drie de vensters dezelfde afbeelding laten zien, het bedrag hangt van de afbeelding af. Als twee aanliggende vensters dezelfde afbeelding laten zien wint u ook, behalve als het twee citroenen zijn.

Programma-opmerkingen

Het programma ontwerpt een fruitmachine met drie onafhankelijk van elkaar draaiende cilinders. Elk venster is opgebouwd uit 20 karakters (4 breed, 5 hoog) en elk venster kan



een verschillend plaatje laten zien - citroen, bel, kers enz
Daar er slechts 21 grafische karakters mogelijk zijn, beperkt dat meteen onze mogelijkheden. Maar gelukkig heeft u niet alle 60 karakters gelijktijdig nodig.

Nadat een grafisch karakter eenmaal op het scherm is gezet houden de geheugenadressen, corresponderende met de positie van dat karakter, hun inhoud totdat er iets anders wordt ingezet. Wat u ondertussen met het grafisch karakter doet dat u geprogrammeerd had, heeft geen invloed op het scherm.
U veronderstelt bijvoorbeeld dat u A gedefinieerd had als BIN 01010101, BIN 10101010, BIN 01010101 enz., en dat u de opdracht

PRINT AT 0,0; "A"

heeft gegeven. De acht geheugenadressen corresponderende met (0,0) herinneren zich dat als BIN 01010101, BIN 10101010 enz., en niet als 'A'. U kunt dus A opnieuw programmeren zonder dat het gevolgen heeft voor het scherm. Dit geldt echter niet voor alle computers.
Dit programma tekent dus vormen, herprogrammeert de karakters, tekent dan weer nieuwe vormen enz.

Programma-opbouw

100-200	Definieer variabelen
950-1810	Teken fruitmachine en controleer krediet speler
2000-2230	Hoofdlus voor elk venster voor elk van de 3 tot 6 plaatjes kies een plaatje voor elk van de 5 regels herprogrammeer karakterset vul elk blok op scherm volgende regel volgend plaatje herprogrammeer karakterset volgend venster
2500-2560	Als winst, bereken winst

Het programma haalt alle informatie uit DATA-statements, zelfs de inkt- en papierkleur die gebruikt moeten worden om het volgende blok te tekenen.

16K- en 48K-versie
Zoals het programma nu is past het precies in een 16K-machine.
Het is niet bepaald efficient om informatie in DATA-statements op te slaan (zie de aanwij-

zingen bij Meteorenstorm), en er zijn ook geen pogingen ondernomen om het geheugengebruik te minimaliseren.
Deze versie heeft 3 verschillende plaatjes: kers, citroen en bel.
Als u het programma wilt uitbreiden voor 48K dan zult u uw eigen afbeeldingen moeten ontwerpen. Bestudeert u hoe dit programma de afbeeldingen in DATA-statements heeft opgezet (bijv. de kers in regel 8100 - 8153).
De door u ontworpen afbeeldingen moeten in DATA-statements gezet worden, te beginnen met een regelnummer 100 groter dan het begin van de laatste afbeelding (in dit geval de bel op regel 8310). Denkt u er wel aan dat elke afbeelding uit exact 160 DATA moet bestaan.
De volgende wijzigingen moeten in het 48K-programma worden aangebracht:

* vanaf regel 9000 staat er voor elke afbeelding een DATA-statement met 10 DATA voor INK en PAPER;
* regel 2040 kiest een van afbeeldingen:
  LET plaat = 1 + INT (3*RND)
  verander het getal 3 in het aantal aanwezige afbeeldingen

Door gebruik te maken van één van de geheugenruimte-besparende technieken die elders in dit boek worden beschreven, moet het mogelijk zijn om 5 afbeeldingen in een 16K-machine te krijgen. Probeert u het eens!

```
Je hebt 9 gulden over
Toets <ENTER> voor start
```

Programma

```
  10 REM fruit machine
 100 LET vulhires=5300
 110 LET tekenblok=5100
 120 LET eerst=5
 130 LET resethires=5500
 140 LET elsvenster=5700
 150 LET geld=10
 160 DIM r(3)
 170 DIM s(4)
 180 RESTORE 7000
 190 FOR k=1 TO 4: READ s(k): NEXT k
 200 INK 7: PAPER 5: BORDER 5: CLS
 950 REM teken fruitmachine
 980 LET c$="GRA SHI 8 SH1 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 GRU"
```

```
1000 PRINT AT 1,5;"GRA 4 GRU";c$;"GRA SHI 7 GRU"
1010 PRINT AT 2,4;"GRA 4 SHI 8 GRU";c$;"GRA SHI 8 SHI 7 GRU
"
1020 PRINT AT 3,3;"GRA 4 SHI 8 SHI 8 GRU";c$;'GRA SHI 8 SHI
 8 SHI 7 ## 4 GRU"
1030 PRINT AT 4,3;"GRA SHI 8 SHI 8 SHI 8 GRU';c$;'GRA SHI 8
 SHI 8 SHI 8 ## SHI 8 SHI 5 GRU"
1040 PRINT AT 5,3;"GRA SHI 8 SHI 8 SHI 8 GRU";c$;'GRA SHI 8
 SHI 8 SHI 8 ## SHI 8 SHI 5 GRU"
1050 PRINT AT 6,3;"GRA SHI 8 SHI 8 #### SHI 8 SHI 8 SHI 8 #
### SHI 8 SHI 8 SHI 8 #### SHI 8 SHI 8 ## SHI 8 SHI 5 GRU"
1060 PRINT AT 7,3;"GRA SHI 8 SHI 8 #### SHI 8 SHI 8 SHI 8 #
### SHI 8 SHI 8 SHI 8 #### SHI 8 SHI 8 ## SHI 8 SHI 5 GRU"
1070 PRINT AT 8,3;"GRA SHI 8 SHI 8 #### SHI 8 SHI 8 SHI 8 #
### SHI 8 SHI 8 SHI 8 #### SHI 8 SHI 8 ## 5 GRU"
1080 PRINT AT 9,3;"GRA SHI 8 SHI 8 #### SHI 8 SHI 8 SHI 8 #
### SHI 8 SHI 8 SHI 8 #### SHI 8 SHI 8 ## 5 GRU"
1090 PRINT AT 10,3;"GRA SHI 8 SHI 8 #### SHI 8 SHI 8 SHI 8
#### SHI 8 SHI 8 SHI 8 #### SHI 8 SHI 8 ## 5 GRU'
1100 PRINT AT 11,3;"GRA SHI 8 SHI 8 GRU';c$;"GRA SHI 8 SHI
8 SHI 8 SHI 8 SHI 3 SHI 3 SHI 2 GRU"
1110 PRINT AT 12,3;"GRA SHI 8 SHI 8 GRU";c$;"GRA SHI 8 SHI
8 SHI 8 SHI 8 GRU"
1120 PRINT AT 13,3;"GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 EXT SHI 0 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SH
I 8 SHI 8 EXT SHI 7 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 EXT SHI 0"
1130 PRINT AT 14,3;"GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 EXT SHI 0 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SH
I 8 SHI 8 EXT SHI 7 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 EXT SHI 0"
1140 PRINT AT 15,3;"GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 EXT SHI 0 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SH
I 8 SHI 8 SHI 8 EXT SHI 7 GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 EXT SHI 0"
1150 PRINT AT 16,3;"SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI
 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SH
I 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 GRU"
1160 PRINT AT 17,3;"GRA SHI 8 SHI 8 GRU";c$;"GRA SHI 8 SHI
8 SHI 8 SHI 8 GRU"
1500 GO SUB clsvenster
1750 IF geld>0 THEN  GO TO 1780
1760 PRINT  INK 0;AT 20,0;'Sorry#-#je#hebt#geen#geld#meer"
1770 STOP
1780 PRINT  INK 0;AT 20,0;'Je#hebt#";geld;" #gulden#over";AT
 21,0;'Toets#ENTER #voor#start"
1790 INPUT k$
1800 PRINT AT 20,0;"########################################
##########################'
1810 LET geld=geld-.5
1820 PRINT  INK 0; PAPER 0;AT 13,13;"##";AT 14,12;"####';AT
 15,10;'########"
1830 GO SUB clsvenster
2000 REM display alle vensters
2010 FOR w=1 TO 3
```

```
2020 FOR d=1 TO 3+3*RND
2030 LET oudplaat=plaat
2040 LET plaat=1+INT (3*RND)
2050 IF plaat=oudplaat THEN  GO TO 2040
2060 FOR l=5 TO 1 STEP -1
2070 GO SUB vulhires
2080 LET p=0
2090 LET patroon=plaat
2100 FOR b=1 TO 5
2110 LET p=p+1
2120 GO SUB tekenblok
2130 NEXT b
2140 LET patroon=oudplaat
2150 FOR b=1 TO l-1
2160 LET p=p+1
2170 GO SUB tekenblok
2180 NEXT b
2190 NEXT l
2200 NEXT d
2210 LET r(w)=plaat
2220 GO SUB resethires
2230 NEXT w
2500 IF r(1)<>r(2) AND r(2)<>r(3) THEN  GO TO 1750
2510 IF r(2)=2 THEN  GO TO 1750
2520 LET win=1
2530 IF r(1)=r(2) AND r(2)=r(3) THEN  LET win=6*(r(1))
2540 PRINT  PAPER 0;AT 13,13;"JE";AT 14,12;"WINT";AT 15,10;
win;' GULDEN"
2550 LET geld=geld+win
2560 GO TO 1750
5100 REM kies goede kleuren voor blok
5110 RESTORE 9000+100*patroon
5120 FOR c=1 TO b-1
5130 READ ink,paper
5140 NEXT c
5150 READ ink,paper
5200 REM print een blok
5210 PRINT AT eerst+p,7*w-2;
5220 FOR c=1 TO 4
5230 PRINT  INK ink; PAPER paper;CHR$ (144+4*b-5+c);
5240 NEXT c
5250 RETURN
5300 REM definieer high resolution karakters opnieuw
5310 RESTORE 8000+100*plaat+10*l
5320 FOR h= USR "a"+32*l-32 TO USR "a"+32*l-1
5330 READ byte: POKE h,byte
5340 NEXT h
5350 RETURN
5500 REM definieer hires als hers
5510 LET patroon=1
5520 LET plaat=1
5530 LET oudplaat=1
5540 FOR l=5 TO 1 STEP -1
5550 GO SUB vulhires
5590 NEXT l
```

```
5600 RETURN
5700 REM vul alle vensters met kers
5710 LET hendel=1: GO SUB resethires: LET hendel=0
5720 FOR w=1 TO 3
5730 LET d=1
5740 FOR b=1 TO 5
5750 LET p=b
5760 GO SUB tekenblok
5770 NEXT b
5810 NEXT w
5820 RETURN
7000 DATA 15,3,12,7
8000 REM data voor high resolution plaatjes
8100 REM kers
8110 DATA 0,0,0,0,0,0,0,0
8111 DATA 0,0,0,0,0,0,0,0
8112 DATA 0,6,6,14,30,30,62,126
8113 DATA 0,0,0,0,0,0,0,0
8120 DATA 0,0,0,0,0,0,0,0
8121 DATA 0,0,1,1,1,2,2,2
8122 DATA 127,191,31,3,0,0,0,0
8123 DATA 0,0,0,128,128,0,0,0
8130 DATA 0,0,0,0,0,0,0,0
8131 DATA 4,4,4,4,4,4,2,2
8132 DATA 0,0,0,0,0,0,0,0
8133 DATA 0,0,0,0,0,0,0,0
8140 DATA 0,0,0,0,1,1,1,1
8141 DATA 29,60,126,255,255,255,255,255
8142 DATA 224,240,120,124,252,254,254,254
8143 DATA 0,0,0,0,0,0,0,0
8150 DATA 1,1,0,0,0,0,0,0
8151 DATA 255,255,255,127,63,31,7,0
8152 DATA 254,254,252,252,248,224,128,0
8153 DATA 0,0,0,0,0,0,0,0
8200 REM citroen
8210 DATA 0,0,0,0,0,0,0,0
8211 DATA 0,0,0,0,0,0,0,1
8212 DATA 0,0,0,0,0,0,127,255
8213 DATA 0,0,0,0,0,0,16,188
8220 DATA 0,0,0,0,1,1,3,7
8221 DATA 7,31,127,255,254,248,240,224
8222 DATA 255,255,255,255,15,15,31,63
8223 DATA 254,254,126,60,196,246,254,254
8230 DATA 7,15,31,31,63,63,63,63
8231 DATA 192,193,199,255,255,255,255,255
8232 DATA 127,255,255,255,255,255,255,255
8233 DATA 254,254,254,254,254,252,252,252
8240 DATA 63,63,63,63,63,63,31,31
8241 DATA 255,255,255,255,255,255,255,255
8242 DATA 255,255,255,255,255,255,255,255
8243 DATA 248,248,240,224,224,192,128,0
8250 DATA 63,63,63,63,31,0,0,0
8251 DATA 255,255,255,255,0,0,0,0
8252 DATA 254,248,224,128,0,0,0,0
8253 DATA 0,0,0,0,0,0,0,0
```

```
8300 REM bel
8310 DATA 0,0,0,0,0,0,0,0
8311 DATA 0,0,3,14,8,8,12,6
8312 DATA 0,0,192,112,16,16,48,96
8313 DATA 0,0,0,0,0,0,0,0
8320 DATA 0,0,0,0,0,0,0,0
8321 DATA 3,15,63,127,127,255,255,255
8322 DATA 192,240,252,254,254,255,255,255
8323 DATA 0,0,0,0,0,0,0,0
8330 DATA 0,1,1,1,1,1,1,1
8331 DATA 255,255,255,255,255,255,255,255
8332 DATA 255,255,255,255,255,255,255,255
8333 DATA 0,128,128,128,128,128,128,128
8340 DATA 1,3,3,3,3,7,14,56
8341 DATA 255,255,255,255,127,127,127,255
8342 DATA 255,255,255,255,255,255,255,255
8343 DATA 128,192,192,192,192,224,240,252
8350 DATA 31,3,0,0,0,0,0,0
8351 DATA 255,255,3,3,1,0,0,0
8352 DATA 255,255,128,128,0,0,0,0
8353 DATA 252,192,0,0,0,0,0,0
9000 REM INK en PAPER data voor elk blok
9100 DATA 4,1,4,1,4,1,2,1,2,1
9200 DATA 6,4,6,4,6,4,6,4,6,4
9300 DATA 7,2,7,2,7,2,7,2,7,2
```

4. Wiskundige programma's

4.1 'Bubble'-sorteerroutine
(Copyright Beam Software)

Allerlei soorten mensen zijn geïnteresseerd in het werken met computers, maar in de wiskunde heeft het woord sorteren een bepaalde betekenis. Er bestaan namelijk veel soorten sorteren.
Een sortering is een handeling waarmee u orde kunt scheppen in een chaos, vaak wordt de sorteerroutine toegepast bij een groot aantal namen of een reeks getallen.

Het programma dat hier volgt is zowel een demonstratie van de handelingen die bij een sorteerroutine komen kijken, als een basis voor een compacte subroutine die u in uw eigen programma's kunt gebruiken - bijvoorbeeld om een lijst namen en adressen te sorteren, of een hand kaarten in een bridgespel.
De demonstratie die bij dit sorteerprogramma hoort is grafisch gezien een lust voor het oog en tegelijkertijd langzaam genoeg om de theorie van het sorteren goed te kunnen volgen.

De 'bubble'-sorteerroutine
Het eerste programma is een traditionele 'bubble'-sorteerroutine. Het programma vergelijkt twee opeenvolgende getallen in een reeks en als het eerste getal groter is dan het tweede dan verwisselt het programma ze.
Als u dit programma laat 'lopen' dan realiseert u zich al snel waarom het een 'bubble'-sorteerroutine heet - de lage getallen lijken naar boven te 'bubbelen' en de hoge naar beneden.
Het programma blijft maar twee opeenvolgende getallen verwisselen terwijl de reeks getallen steeds beter gesorteerd wordt, totdat er niets meer gesorteerd hoeft te worden.
Het programma voert de sorteerroutine heel leuk uit en u kunt zien hoe hard de bulldozer moet werken om de rotzooi op te ruimen. U zult zich dan ook al snel realiseren dat er een betere methode moet zijn om dingen te sorteren.

Aangepaste 'bubble'-sorteerroutine
Het tweede programma is een aangepaste 'bubble'-routine. Het programma is nu niet langer bezig om alleen maar opeenvolgende getallen te bekijken, maar is ook bereid om veel verder in de reeks te gaan zoeken.
Dit betekent dat het aantal keren dat getallen van volgorde moeten verwisselen tot ongeveer de helft kan worden teruggebracht vergeleken met de traditionele sorteerroutine.
U kunt óf het tweede programma intikken van de uitdraai, of, als u het eerste programma al heeft, regel 320, 330, 340, 500 en 510 veranderen.

Sorteer-subroutine
U kunt sorteerroutines behalve voor demonstratiedoeleinden natuurlijk ook in uw eigen programma's gebruiken. Het derde programma is een uitdraai van de aangepaste 'bub-

ble'-routine in subroutinevorm (regel 8000-8030) die u in uw eigen programma's kunt gebruiken
Deze subroutine gaat er van uit dat de getallen die gesorteerd moeten worden in een array p(n) staan, waarbij n het aantal getallen is dat gesorteerd moet worden
Programma 3 kan uit programma 2 verkregen worden door regels 110-130 te verwijderen en alle regels groter dan 280. Voegt u daarna de nieuwe regels 280-300, 1000-1040 en 8000-8030 toe
Zoals het programma laat zien kunnen 15 getallen in ongeveer 3 seconden gesorteerd worden, inclusief de tijd die nodig is om de 15 getallen twee keer op het scherm te zetten

Programma

```
  10 REM bubble1
 100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
 110 RESTORE 600
 120 FOR i=USR "a" TO USR "d"-1
 130 READ x: POKE i,x: NEXT i
 140 LET n=15
 150 DIM p(n)
 160 PRINT "Wil#je#zelf#de#getallen#kiezen##(j/n)?"
 170 LET b$=INKEY$: IF b$="" THEN  GO TO 170
 180 IF b$="n" OR b$="N" THEN  GO TO 240
 190 PRINT '"Tik#de#15#getallen#in#die#je####gesorteerd#wilt#hebben"
 200 FOR y=1 TO n
 210 INPUT p(y): PRINT p(y)
 220 NEXT y
 230 GO TO 270
 240 FOR x=1 TO n
 250 LET p(x)=15*RND+1
 260 NEXT x
 270 CLS
 280 FOR x=1 TO n
 290 PRINT AT x,0;p(x)
 300 NEXT x
 310 LET m=18
 320 FOR i=n-1 TO 1 STEP -1
 330 FOR j=1 TO i
 340 LET l=j+1
 350 IF p(j)<=p(l) THEN  GO TO 500
 360 LET b$="##########"( TO LEN STR$ p(j)): PRINT AT j,0; OVER 1; PAPER 5;b$
 370 IF l=m THEN  GO TO 390
 380 FOR x=m TO l STEP SGN (l-m): PRINT  INK 2;AT x,18;"GRA ABC GRU";AT x+SGN (m-l),18;"###": NEXT x
 390 LET y=LEN STR$ p(l): FOR x=17 TO y STEP -1: PRINT  INK 2;AT l,x;"GRA ABC GRU#": NEXT x
 400 FOR k=0 TO 10: PRINT AT l,k;"#";p(l); INK 2;"GRA ABC GRU": NEXT k
 410 FOR x=1 TO j STEP -1: PRINT AT x,11;p(l); INK 2;"GRA ABC GRU";AT x+1,11;"##############": NEXT x
 420 PRINT  OVER 1;AT j,0;p(j)
```

```
 430 FOR x=j TO 1: PRINT  OVER 1;AT x,0;p(j);AT x,0;p(j): N
EXT x: PRINT AT 1,0;p(j)
 440 FOR x=10 TO 0 STEP -1: PRINT AT j,x;p(1); INK 2;"GRA A
BC GRU";"#": NEXT x
 450 FOR x=y TO 17: PRINT AT j,x; INK 2;"#GRA ABC GRU": NEX
T x
 460 LET m=j
 470 LET t=p(1)
 480 LET p(1)=p(j)
 490 LET p(j)=t
 500 NEXT j
 510 NEXT i
 520 FOR x=m TO 18: PRINT AT x,18; INK 2;"GRA ABC GRU";AT x
-1,18;"###": NEXT x
 530 PRINT AT 21,0;"Wil#je#nog#een#sortering#(j/n)#?"
 540 LET b$=INKEY$: IF b$="" THEN  GO TO 540
 550 IF b$="j" OR b$="J" THEN  RUN 100
 560 IF b$="y" OR b$="Y" THEN  RUN 100
 600 DATA 0,0,2,6,15,30,63,254
 610 DATA 0,0,63,127,241,100,238,4
 620 DATA 63,255,241,228,158,223,14,4
```

Programma

```
  10 REM bubble2
 100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
 110 RESTORE 600
 120 FOR i=USR "a" TO USR "d"-1
 130 READ x: POKE i,x: NEXT i
 140 LET n=15
 150 DIM p(n)
 160 PRINT "Wil#je#zelf#de#getallen#kiezen##(j/n)?"
 170 LET b$=INKEY$: IF b$="" THEN  GO TO 170
 180 IF b$="n" OR b$="N" THEN  GO TO 240
 190 PRINT '"Tik#de#15#getallen#in#die#je###gesorteerd#wil
t#hebben"
 200 FOR y=1 TO n
 210 INPUT p(y): PRINT p(y)
 220 NEXT y
 230 GO TO 270
 240 FOR x=1 TO n
 250 LET p(x)=15*RND+1
 260 NEXT x
 270 CLS
 280 FOR y=1 TO n
 290 PRINT AT y,0;p(y)
 300 NEXT x
 310 LET m=18
 320 FOR j=1 TO n-1
 330 FOR i=j+1 TO n
 340 LET l=n+j-i+1
 350 IF p(j)<=p(l) THEN  GO TO 500
```

```
 360 LET b$="##########"( TO LEN STR$ p(j)): PRINT AT j,0; O
VER 1; PAPER 5;b$
 370 IF 1=m THEN  GO TO 390
 380 FOR x=m TO 1 STEP SGN (1-m): PRINT  INK 2;AT x,18;"GRA
 ABC GRU";AT x+SGN (m-1),18;"###": NEXT x
 390 LET y=LEN STR$ p(1): FOR x=17 TO y STEP -1: PRINT  INK
 2;AT 1,x;'GRA ABC GRU#": NEXT x
 400 FOR x=0 TO 10: PRINT AT 1,x;"#";p(1); INK 2;"GRA ABC G
RU": NEXT x
 410 FOR x=1 TO j STEP -1: PRINT AT x,11;p(1); INK 2;"GRA A
BC GRU";AT x+1,11;"##############": NEXT x
 420 PRINT  OVER 1;AT j,0;p(j)
 430 FOR x=j TO 1: PRINT  OVER 1;AT x,0;p(j);AT x,0;p(j): N
EXT x: PRINT AT 1,0;p(j)
 440 FOR x=10 TO 0 STEP -1: PRINT AT j,x;p(1); INK 2;'GRA A
BC GRU";"#": NEXT x
 450 FOR x=y TO 17: PRINT AT j,x; INK 2;'#GRA ABC GRU": NEX
T x
 460 LET m=j
 470 LET t=p(1)
 480 LET p(1)=p(j)
 490 LET p(j)=t
 500 NEXT i
 510 NEXT j
 520 FOR x=m TO 18: PRINT AT x,18; INK 2;"GRA ABC GRU";AT x
-1,18;"###": NEXT x
 530 PRINT AT 21,0;"Wil#je#nog#een#sortering#(j/n)#?"
 540 LET b$=INKEY$: IF b$="" THEN  GO TO 540
 550 IF b$="j" OR b$="J" THEN  RUN 100
 560 IF b$="y" OR b$="Y" THEN  RUN 100
 600 DATA 0,0,2,6,15,30,63,254
 610 DATA 0,0,63,127,241,100,238,4
 620 DATA 63,255,241,228,158,223,14,4
```

Programma

```
  10 REM bubble3
 100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
 140 LET n=15
 150 DIM p(n)
 160 PRINT "Wil#je#zelf#de#getallen#kiezen##(j/n)?"
 170 LET b$=INKEY$: IF b$="" THEN  GO TO 170
 180 IF b$="n" OR b$="N" THEN  GO TO 240
 190 PRINT '"Tik#de#15#getallen#in#die#je####gesorteerd#wil
t#hebben"
 200 FOR y=1 TO n
 210 INPUT p(y): PRINT p(y)
 220 NEXT y
 230 GO TO 270
 240 FOR x=1 TO n
 250 LET p(x)=15*RND+1
 260 NEXT x
 270 CLS
```

```
 280 GO SUB 1000: GO SUB 8000
 290 GO SUB 1000
 300 STOP
1010 FOR x=1 TO n
1020 PRINT AT x,0;p(x)
1030 NEXT x
1040 RETURN
8000 REM sorteer subroutine
8010 FOR j=1 TO n-1: FOR i=j+1 TO n: LET l=n+j-i+1
8020 IF p(l)<=p(j) THEN  LET t=p(l): LET p(l)=p(j): LET p(j
)=t
8030 NEXT i: NEXT j: RETURN
```

4.2 Stelsel van vergelijkingen
(Copyright Beam Software)

Dit programma lost een stelsel van vergelijkingen op van het type

ax + by + c = 0

Dit is de algemene vorm van een vergelijking van een rechte lijn en de 'oplossing' van twee dergelijke vergelijkingen geeft de coördinaten van het punt waar de lijnen elkaar snijden.

Het programma vraagt voor elk van de twee vergelijkingen de waarde op voor a, b en c, waarna de waarden voor x en y berekend worden voor het snijpunt.
Als de lijnen elkaar niet snijden - ze lopen dan parallel - dan wordt dit door het programma aangegeven.

Programma-opbouw
Het programmagedeelte dat voor de berekening zorgt is maar vrij kort, het grootste gedeelte van het programma bestaat uit het opvragen van de gewenste informatie bij de gebruiker en uit het op het scherm zetten van de resultaten en de grafiek.

In regel 830 wordt de determinant D berekend. Als D gelijk is aan 0, dan bestaat er geen snijpunt.

Regel 900-990 bepaalt in welk kwadrant de lijnen elkaar snijden, en in welke richting de grafiek uitgezet moet worden.

Programma

```
100 REM stelsel
101 REM
110 REM van vergelijkingen
120 BORDER 7: INK 1: PAPER 7: FLASH 0: INVERSE 0: CLS
130 PRINT AT 0,2; INVERSE 1;"STELSEL#VAN#VERGELIJKINGEN"
```

```
 140 PRINT AT 2,0; INK 2; INVERSE 1;"a1X+b1Y+c1=0"
 150 PRINT AT 2,16; PAPER 6; INK 0;"a2X+b2Y+c2=0"
 160 PRINT AT 4,0;"a1=";AT 4,16;"a2=";AT 5,0;"b1=";AT 5,16;
"b2=";AT 6,0;"c1=";AT 6,16;"c2="
 170 DIM C(6): REM coeficient array
 180 DIM S$(7)
 190 DIM X$(7)
 200 DIM Y$(7)
 500 REM
 510 REM invoer vergelijking 1
 520 FOR I=1 TO 3
 530 PRINT AT I+3,3; FLASH 1;">"; FLASH 0; INK 2; INVERSE 1
;"##############"
 540 INPUT S$: LET C(I)=VAL S$
 550 PRINT AT I+3,3;"###############";AT I+3,4;ABS C(I)
 560 IF SGN C(I)=-1 THEN  PRINT AT I+3,3;"-"
 570 NEXT I
 600 REM
 610 REM invoer vergelijking 2
 620 FOR I=4 TO 6
 630 PRINT AT I,19; FLASH 1;" "; FLASH 0; PAPER 6;"########
####"
 640 INPUT S$: LET C(I)=VAL S$
 650 PRINT AT I,19;"###############";AT I,20;ABS C(I)
 660 IF SGN C(I)=-1 THEN  PRINT AT I,19;"-"
 670 NEXT I
 800 REM
 810 REM bereken  snijpunt
 820 PRINT AT 7,0;
 830 LET D=C(2)*C(4)-C(1)*C(5)
 840 IF D=0 THEN  PRINT  FLASH 1;"GEEN#SNIJPUNT": PAUSE 100
: RUN
 850 LET A=(C(3)*C(5)-C(2)*C(6))/D
 860 LET X$=STR$ A: LET A=VAL X$
 870 LET B=(C(1)*C(6)-C(3)*C(4))/D
 880 LET Y$=STR$ B: LET B=VAL Y$
 890 PRINT  INVERSE 1;"SNIJPUNT:"; INVERSE 0;"#X=#";X$;"##Y
=#";Y$
 900 REM
 910 REM teken assen
 920 IF A>=0 AND B>=0 THEN  LET ox=72: LET oy=0: PRINT AT 2
1,8;"O";AT 8,8;"Y";AT 21,23;"X"
 930 IF A>=0 AND B<0 THEN  LET ox=72: LET oy=111: PRINT AT
8,8;"O";AT 21,8;"Y";AT 8,23;"X"
 940 IF A<0 AND B>=0 THEN  LET ox=183: LET oy=0: PRINT AT 2
1,23;"O";AT 21,8;"X";AT 8,23;"Y"
 950 IF A<0 AND B<0 THEN  LET ox=183: LET oy=111: PRINT AT
8,23;"O";AT 8,8;"X";AT 21,23;"Y"
 960 LET dx=(SGN A>=0)-(SGN A<0)
 970 LET dy=(SGN B>=0)-(SGN B<0)
 980 PLOT ox,oy: DRAW  INK 0;dx*111,0: PLOT ox,oy: DRAW  IN
K 0;0,dy*111
 990 GO SUB 1500: REM teken lijnen
1000 INPUT "Nieuwe#vergelijkingen#(j/n)?#";k$
1100 IF k$="n" OR k$="N" THEN  STOP
```

```
1110 IF k$="j" OR k$="J" THEN  RUN
1120 GO TO 1000
1500 REM
1510 LET rx=ABS (55/A): LET ry=ABS (55/B)
1520 REM teken lijn 1
1530 PRINT AT 2,0; INK 2; OVER 1; FLASH 1;"##############"
1540 IF C(1)=0 THEN  LET ix1=0: LET iy1=-C(3)/C(2): LET ix2
=2*A: LET iy2=iy1: GO TO 1620
1550 IF C(2)=0 THEN  LET iy1=0: LET ix1=-C(3)/C(1): LET iy2
=2*B: LET ix2=ix1: GO TO 1620
1560 LET ix1=2*A: LET iy1=-(C(3)+C(1)*ix1)/C(2)
1570 IF dy*iy1<0 THEN  LET iy1=0: GO TO 1590
1580 IF (ABS iy1-ABS (2*B))>0 THEN  LET iy1=2*B
1590 LET ix2=0: LET iy2=-C(3)/C(2)
1600 IF dy*iy2<0 THEN  LET iy2=0: GO TO 1620
1610 IF (ABS iy2-ABS (2*B))>0 THEN  LET iy2=2*B
1620 PLOT ox+rx*ix1,oy+ry*iy1
1630 DRAW (ix2-ix1)*rx,(iy2-iy1)*ry
1640 PAUSE 100
1650 PRINT AT 2,0; INK 2; INVERSE 1;"a1X+b1Y+c1=0"
1800 REM
1810 REM teken lijn 2
1820 PRINT AT 2,16; OVER 1; FLASH 1; PAPER 6; INK 0;"#######
#######"
1830 PLOT  OVER 1;ox+rx*ix1,oy+ry*iy1
1840 DRAW  OVER 1;(ix2-ix1)*rx,(iy2-iy1)*ry
1850 IF C(4)=0 THEN  LET ix3=0: LET iy3=-C(6)/C(5): LET ix4
=2*A: LET iy4=iy3: GO TO 1930
1860 IF C(5)=0 THEN  LET iy3=0: LET ix3=-C(6)/C(4): LET iy4
=2*B: LET ix4=ix3: GO TO 1930
1870 LET ix3=2*A: LET iy3=-(C(6)+C(4)*ix3)/C(5)
1880 IF dy*iy3<0 THEN  LET iy3=0: GO TO 1900
1890 IF (ABS iy3-ABS (2*B))>0 THEN  LET iy3=2*B
1900 LET ix4=0: LET iy4=-C(6)/C(5)
1910 IF dy*iy4<0 THEN  LET iy4=0: GO TO 1930
1920 IF (ABS iy4-ABS (2*B))>0 THEN  LET iy4=2*B
1930 PLOT ox+ix3*rx,oy+iy3*ry
1940 DRAW (ix4-ix3)*rx,(iy4-iy3)*ry
1950 PAUSE 100
1960 PRINT AT 2,16; INK 0; PAPER 6;"a2X+b2Y+c2=0"
1970 PLOT ox+ix1*rx,oy+iy1*ry
1980 DRAW (ix2-ix1)*rx,(iy2-iy1)*ry
1990 RETURN
```

5. Arcade-spelletjes

5.1 Vlucht uit de ruimte
(Copyright Clifford Ramshaw)

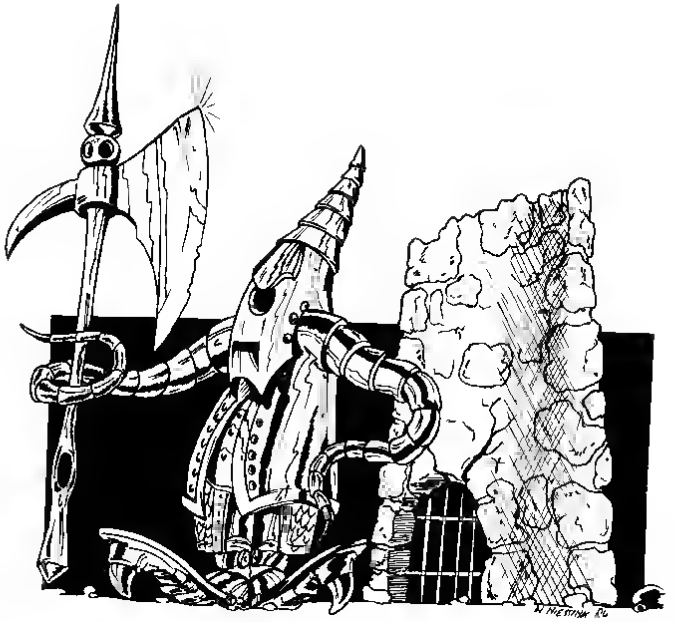
Beschrijving
U bent gevangen genomen door de Alien en u zit vast in een toren die bewaakt wordt door de Alien-robot. U moet opboksen tegen de zwaartekracht en de geweldige trekkrachten van de Alien die u de toren proberen in te sleuren.
Erger nog, als de robot merkt dat u ontsnapt dan wordt de bewegende laser-zuigstraal op u gericht. Wordt u geraakt, dan wordt u de toren ingezogen en weer gevangen gezet.
Veel geluk met uw vluchtpogingen!

Gebruik van het programma
Tik RUN + ENTER in om het programma op te starten.
Met de 'w'-toets kunt u omhoog bewegen, met de 'd'-toets vooruit.
Om te kunnen ontsnappen moet u eerst boven de gevangeniscel uit zien te komen. Als u daarna niet voortdurend vooruit blijft gaan, wordt u teruggezogen in de toren. Als u de 'w'-toets niet gebruikt valt u ook weer terug in de cel.
Telkens als u neergeschoten bent, of met succes gevlucht bent, stopt het programma. Het indrukken van welke toets dan ook geeft u een nieuw spel.

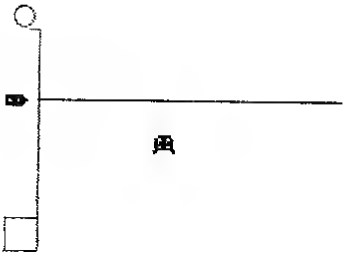
Programma-opbouw
Dit programma gebruikt speciale vrij programmeerbare karakters om de Alien-robot en uw ruimteschip te tekenen.
Een vrij nauwkeurige klok wordt bijgehouden door de interne beeldteller uit te lezen. Deze klok berekent hoe lang het duurt voordat u weer gevangen genomen bent, of, als u geluk heeft, ontsnapt bent.

```
Initialisatie
  zet schermvariabelen op
  zet VPK's op

Hoofdlus:
  teken toren en grenslijn veilige zone
  zet variabele op
L1: sla oude positie op
  bereken nieuwe positie
  beweging vooruit?
    nee, produceer geluid
  teken wisselende vorm ruimteschip
  grenslijn bereikt?
    ja, bereid melding voor
    ga naar T1;
  wordt er geschoten?
    nee en RND < 9, ga naar L1
```



```
    schietvlag aan?
      nee, zet vlag aan
      bereken uiteindelijke positie robot
    veeg oude robot uit
    breng robot naar volgende positie
    teken robot
    uiteindelijke positie bereikt?
      nee, ga naar L1:
    vuur laserstraal af
    ruimteschip geraakt?
      nee, ga naar L1:
    voer terugtrekroutine uit
    bereid melding voor
T1: bereken verstreken tijd
    zet uiteindelijke melding op scherm
```



Opmerkingen

De berekende verstreken tijd is vrij nauwkeurig en wordt alleen beïnvloed door de tijd die nodig is om geluid te produceren

Programma

```
 100 REM vlucht uit de ruimte
 120 GO TO 8000
 500 REM
 510 REM definieer vorm schip
 520 FOR I=USR "A" TO (USR "J"+7)
 550 READ m: POKE I,m
 570 NEXT I
 580 DATA 0,0,0,30,30,49,61,61,0,0,0,60,60,198,246,246,63,4
9,49,63,14,14,2,0,254,198,198,254,56,56,32,0
 590 DATA 0,0,0,15,15,49,49,55,0,0,0,248,248,198,198,222,55
,63,49,49,63,112,96,64,222,254,198,198,254,7,3,1
 600 DATA 255,255,255,207,207,255,255,255,240,248,252,63,63
,252,248,240
1000 REM
1010 REM teken toren
1020 CIRCLE  INK 0;12,167,7
1030 INK 0: PLOT 16,158: DRAW 7,0: DRAW 0,-158: DRAW -23,0:
 DRAW 0,23: DRAW 23,0
1040 PLOT 255,23: DRAW 0,-23
1050 FOR I=23672 TO 23674: REM inititialiseer klok
1060 POKE I,0
1070 NEXT I
1080 RETURN
2000 REM
2010 REM terug
2020 PLOT 24,(21-ng)*8+4: DRAW  INK 2;(D-3)*8-1,0
2030 FOR I=(ID+(D>ID)) TO 0 STEP -1
2050 PRINT AT H,I;'GRA AB GRU#": PRINT AT H+1,I;"GRA CD GRU
#"
2060 BEEP .05,(36+I)
2070 NEXT I
2090 FOR J=RH TO 18
2100 PRINT AT J,0;'##': PRINT AT J+1,0;'##'
2110 PRINT AT J+1,0;"GRA AB GRU": PRINT AT J+2,0;"GRA CD GR
U"
2120 BEEP .05,(24+J)
2130 NEXT J
2140 RETURN
8000 REM
8010 REM hoofd loop
8020 BORDER 6: PAPER 6: INK 0: OVER 0: FLASH 0: CLS
8030 RESTORE : GO SUB 500: REM definieer VPK schip
8040 GO SUB 1000: REM teken toren
8050 LET S=30: LET H=20: LET D=3: LET sh=0: LET og=INT (RND
*17)+2: LET fr=sh
8060 LET OH=H: LET OD=D
8070 LET k$=INKEY$
8080 LET H=H+(H<19)-(k$="w" AND H>1)*1.5: LET IH=INT H
8090 LET D=D+(k$="d" AND H<19)*1.5-(D>3)*.3: LET ID=INT D
8100 IF k$<>"d" THEN  BEEP .0125,-24
8110 IF D>30 THEN  LET D=30
8120 IF sh=0 THEN  GO TO 8170
8130 LET sh=0
```

```
8140 PRINT AT OH,OD;"##": PRINT AT OH+1,OD;"##"
8150 PRINT AT H,D;"GRA EF GRU": PRINT AT H+1,D;"GRA GH GRU"

8160 GO TO 8200
8170 LET sh=1
8180 PRINT AT OH,OD;"##": PRINT AT OH+1,OD;"##"
8190 PRINT AT H,D;"GRA AB GRU": PRINT AT H+1,D;"GRA CD GRU"

8200 IF D=30 THEN  PRINT AT 21,0;"JE#BENT#ONTSNAPT#IN#";: G
O TO 8370
8210 IF fr=0 AND RND<.9 OR H>19 THEN  GO TO 8060
8220 IF fr=1 THEN  GO TO 8270
8230 LET ng=INT (INT (RND*2)+H)-1: LET ng=ng-(ng>18)
8240 IF ng<2 THEN  LET ng=2
8250 LET gm=(ng>og)-(ng<og)
8260 LET fr=1
8270 IF ng=og THEN  GO TO 8310
8280 PRINT AT og,0;"##": LET og=og+gm
8290 PRINT AT og,0; INK 2;"GRA IJ GRU"
8300 GO TO 8060
8310 PLOT 24,(21-ng)*8+4: DRAW  INK 2;214,0: BEEP .2,20
8320 PLOT  OVER 1;24,(21-ng)*8+4: DRAW  INK 2; OVER 1;214,0
: BEEP .1,0
8330 LET RH=(IH+(H>IH)): IF ng<>RH AND ng<>RH+1 OR RH>19 TH
EN  LET fr=0: GO TO 8060
8340 GO SUB 2000
8350 PRINT AT 21,0;"DE#POGING#DUURDE#";
8370 LET T=INT (((65536*PEEK 23674+256*PEEK 23675+PEEK 2367
2)/(50*60))*100)/100
8380 PRINT T;"#MINUTEN"
8390 IF INKEY$="w" OR INKEY$="d" THEN  GO TO 8390
8400 IF INKEY$="" THEN  GO TO 8400
8410 CLS : GO TO 8040
```

5.2 Maanlander

(Copyright Clifford Ramshaw)

Beschrijving

U bent de piloot van een op afstand bestuurde ruimtecapsule die op het ruige maanoppervlak moet landen.
Elke mislukte poging vernielt de capsule en maakt het onmogelijk dezelfde landingsplaats nog een keer te gebruiken. U moet een andere landingsplaats zoeken en een nieuwe capsule gebruiken. Als de landing geslaagd is wordt er een vlag geplant ten teken dat u aanspraak maakt op dat gebied.
De capsule is onderhevig aan een constante trekkracht die hem naar rechts doet afwijken, tegelijkertijd trekt de zwaartekracht van de maan u naar beneden.
Veel geluk!

Gebruik van het programma

Tik RUN + ENTER in om het programma op te starten.
Gebruik de '5'-toets om naar links te bewegen. Met de '7'-toets kunt u hoogte winnen.

De ruimtecapsule beweegt constant vooruit, maar verliest gaandeweg hoogte door de zwaartekracht van de maan.
Voor een geslaagde landing moet u de twee poten van de capsule precies op de BLUE CYAN-landingsplaats neerzetten.

Programma-opbouw
Het programma gebruikt de PLOT en DRAW-instructies om bij elk spel een willekeurig landschap te tekenen.
Speciale vrij programmeerbare karakters worden gebruikt om de capsule te tekenen.
De kleurattributen worden voortdurend uitgelezen om te kijken of de capsule te pletter is geslagen.

```
Initialisatie
  zet VPK's op
I1: zet landingsvlag op 0
  ga naar Hoofdlus
Hoofdlus
  zet scherm variabelen op
  teken landschap en landingsplaats
  initialiseer positie capsule
L1: sla oude positie op
  bereken nieuwe positie
  veeg oude capsule uit
  teken nieuwe positie
  slaat capsule te pletter?
    ja, teken explosie (GOSUB 2000)
    produceer geluid
    ga naar E1
  landing geslaagd?
    nee, ga naar L1:
  FLASH BORDER
  teken vlag
  produceer geluid
E1: PAUZE
  ga naar I1
```

Opmerkingen
Om te vermijden dat de capsule op het scherm knippert mag er niet te veel tijd zitten tussen het uitvegen van de oude capsule en het tekenen van de nieuwe.
Het scherm heeft PAPER WHITE (7) en INK BLACK (0). Te pletter slaan van de capsule wordt getest door de kleurattributen uit te lezen; het landschap heeft attributen INK RED (2) en PAPER WHITE (7) en de landingsplaats INK BLUE (1) en PAPER CYAN (5). Bij een geslaagde landing moeten dus beide attributen van de posities van de capsulepoten 41 als waarde hebben.
Indien u dus de kleurattributen van landschap, landingsplaats of achtergrond wilt wijzigen, dan moet u ook de testwaarden in regel 8100 en 8150 aanpassen.

Programma

```
 100 REM maanlander
 120 RESTORE :  FOR I=1 TO 16: REM initialiseer VPK
 130 LET S=USR (CHR$ (144+I-1))
 140 FOR J=1 TO 8
 150 READ g: POKE (S+J-1),g
 160 NEXT J
 170 NEXT I
 180 DATA 0,0,1,1,7,7,31,25,0,0,128,128,224,224,248,152,53,
107,127,243,96,96,192,192,172,214,254,207,6,6,3,3
 190 DATA 1,1,1,1,1,1,1,1,224,176,140,134,129,135,140,176,1
,1,7,1,7,7,31,25,224,128,224,128,224,224,248,152
 200 DATA 0,0,0,4,5,15,6,15,0,0,0,0,96,192,224,248,27,7,3,6
,0,0,0,0,120,168,112,32,0,0,0,0
 210 DATA 160,67,38,60,24,48,96,97,4,154,244,124,6,4,4,6,19
4,96,32,36,104,120,79,193,130,14,12,36,124,78,195,2
 220 LET ld=0
 230 GO TO 8000
1000 REM
1010 REM teken basis & landschap
1020 LET bx=INT (RND*21): IF bx<11 THEN  GO TO 1020
1030 LET lt=bx*8: LET rt=(bx+4)*8
1040 PRINT AT 21,bx;"GRA SHI 8 GRU": PRINT AT 21,bx+1; PAPE
R 5; INK 1;"GRA SHI 6 6 GRU": PRINT AT 21,bx+3;"GRA SHI 8 G
RU"
1050 LET d1=-1: GO SUB 1500: REM teken linkerlandschap
1060 LET d1=1: GO SUB 1500: REM teken rechterlandschap
1070 RETURN
1500 REM
1510 REM teken landschap
1520 REM invoer d1
1530 LET ly=8+(d1=1)*16
1540 LET lx=((rt+24)*(d1=1))+(lt*(d1=-1))
1550 LET up=255-(d1=-1)*255
1560 IF d1=1 THEN  PLOT rt,8: DRAW  INK 2;24,16
1570 PLOT lx,ly
1580 LET a=RND: LET ry=((a<=.6)-(a>.6))*(INT (RND*48)+1)
1590 LET rx=d1*(INT (RND*16)+1)
1600 IF d1*(lx+rx)>up*d1 THEN  LET rx=up-lx
1610 IF ly+ry>144 OR ly+ry<0 THEN  LET ry=0-ry
1620 LET lx=lx+rx: LET ly=ly+ry
1630 DRAW  INK 2;rx,ry
1640 IF lx<>up THEN  GO TO 1550
1650 RETURN
2000 REM
2010 REM Explosie routine
2020 PRINT AT H,X;"##"
2030 FOR J=1 TO 5
2040 PRINT AT H+1,X;"GRA AB GRU": PRINT AT H+2,X;"GRA CD GR
U"
2050 BEEP .05,-(RND*48)
2060 FOR I=1 TO 10: NEXT I
2070 PRINT AT H+1,X;"GRA IJ GRU": PRINT AT H+2,X;"GRA KL GR
U"
```

```
2080 BEEP .05,-(RND*48)
2090 FOR I=1 TO 10: NEXT I
2100 PRINT AT H+1,X;"GRA MN GRU": PRINT AT H+2,X;"GRA OP GR
U"
2110 BEEP .05,-(RND*48)
2120 FOR I=1 TO 10: NEXT I
2130 NEXT J
2140 FOR i=h+1 TO 20: PRINT AT i-1,x;"##";AT i,x;"GRA DC GR
U";AT i+1,x;"GRA AB GRU": NEXT i
2150 RETURN
8000 REM
8010 REM hoofd controle loop
8020 BORDER 3: INK 0: PAPER 7: OVER 0: FLASH 0: CLS
8030 GO SUB 1000: REM teken land
8040 LET X=0: LET H=X
8050 LET OH=H: LET OX=X
8060 LET X=X+0.5*(X<30)-(INKEY$="5")
8070 LET H=H+0.5-(INKEY$="7")*(H>0)
8080 PRINT AT OH,OX;"##": PRINT AT OH+1,OX;"##"
8090 PRINT AT H,X;"GRA AB GRU": PRINT AT H+1,X;"GRA CD GRU"

8100 LET cr=ATTR (H+2,X)+ATTR (H+2,X+1)
8110 IF cr<58 OR cr=82 OR cr=112 THEN  GO TO 8140
8120 GO SUB 2000: REM explosie
8130 BEEP .5,17: BEEP .5,15: BEEP .25,13: BEEP .25,12: BEEP
 .5,10: BEEP .25,13: BEEP .25,12: GO TO 8210
8140 IF H>=20 THEN  GO TO 8120
8150 IF cr=82 THEN  LET ld=1: GO TO 8170
8160 IF ld=0 THEN  GO TO 8050
8170 FOR I=1 TO 6: PAUSE 25: BORDER 1: NEXT I: BORDER 3
8180 PRINT AT 18,b%+1;"GRA EF GRU": PRINT AT 19,b%+1;"GRA G
H GRU"
8190 BEEP .125,12: BEEP .25,19: PAUSE 10: BEEP .125,12: BEE
P .25,19: PAUSE 10: BEEP .125,12: BEEP .25,19: PAUSE 10: BE
EP .125,12: BEEP .25,19: PAUSE 10
8200 BEEP .25,12: BEEP .125,14: BEEP .125,15: BEEP .125,22:
 BEEP .125,22: BEEP .125,15: BEEP .125,14: BEEP .125,19: BE
EP .125,14: BEEP .25,12
8210 FOR I=1 TO 100: NEXT I: CLS : GO TO 220
```

5.3 Alien-blitz
(Copyright Clifford Ramshaw)

De enige verdediger van het ruimtestation bent u
Plotseling ziet u ze ... boven uw hoofd vliegt een konvooi vijandelijke Aliens. Ze moeten op de hoogte zijn van uw macht want ze blijven veilig buiten het bereik van de laserstraal.
Maar wat gebeurt er nu? Langzaam maakt een van de Aliens zich los uit de vloot en komt in uw richting gedoken. Flappert hij nu met zijn vleugels of is dat een ruimte-illusie?
Weinig tijd om daar over na te denken - er komen raketten naar beneden suizen. U moet hem te pakken krijgen, maar die jongens zijn snel, vlug, stel de laser in werking en haal hem binnen. Die hebben we gehad. Maar wat is dat? Er maakt zich weer een Alien los uit het konvooi...
Met de 'Z'-toets beweegt u de laser naar links, met de 'C'-toets naar rechts, en met de 'M'-toets vuurt u een straal af.

Opmerkingen
Dit programma is een 'arcade'-spel voor de Spectrum gebaseerd op het soort videospelletjes die u overal tegenkomt.
De belangrijkste voorwaarden waar het programma dan ook aan moet voldoen zijn de snelheid en de visuele effecten. Geluid wordt alleen maar gebruikt als het strikt noodzakelijk is, dit zou anders het programma te zeer vertragen.

In elke cyclus wordt een minimum aan berekeningen uitgevoerd om de snelheid te maximaliseren. Er hoeven steeds maar drie variabelen bijgehouden te worden.

- 2 coordinaten van vijand in duikvlucht.
- kolompositie van speler

De hoofdlus is in 2 cycli verdeeld, één voor vleugels omhoog en één voor vleugels omlaag. Hierdoor kunnen speler en vijand sneller reageren. In de cycli zelf hoeven dan namelijk geen beslissingen meer genomen te worden over de te tekenen vormen.
Om de snelheid verder op te voeren worden de kleurattributen niet afzonderlijk gespecificeerd door middel van INK 4 enz., maar worden als controle-codes binnen de PRINT-opdrachten gegeven.

Programma-opbouw

100 - 340	zet vormen op teken konvooi
500 - 630	eerste deel cyclus pas alle posities aan
650 - 790	tweede deel cyclus
1000 - 1440	speler of Alien schiet raak
1500 - 1540	vijandelijk konvooi is vernietigd

Programma

```
  10 REM alien
 100 DATA 14,31,28,24,16,16,16,16,112,248,56,24,8,8,8,8: RE
M vleugels in ruststand -AB
 110 DATA 3,15,63,127,127,127,255,227,192,240,252,254,254,2
54,255,199: REM bovenste helft alien - CD
 120 DATA 227,227,99,99,127,63,7,0,199,199,198,198,254,252,
224,0: REM onderste helft alien - EF
 130 DATA 240,112,56,248,124,62,255,30,15,14,28,31,62,124,2
55,120,7,0,0,0,0,0,0,0,224,0,0,0,0,0,0,0: REM vleugels omho
og -GHIJ
 140 DATA 0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,224,30,255,62,124,2
48,56,112,240,120,255,124,62,31,28,14,15: REM vleugels omla
ag - KLMN
 150 DATA 16,16,16,16,24,28,31,14,8,8,8,8,24,56,248,112: RE
M vallende vleugels - OP
 160 DATA 0,0,12,15,31,63,127,255,24,60,60,255,255,255,231,
129,0,0,48,240,248,252,254,255: REM laser basis -QRS
 170 DATA 66,66,38,27,15,7,3,1,66,66,100,216,240,224,192,12
8: REM vijandelijke raketten - TU
 180 FOR i=USR "GRA A GRU" TO USR "GRA A GRU"+167
 190 READ x: POKE i,x
 200 NEXT i
 210 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
 220 LET v=21
 230 LET s=1
 240 LET g=5
 250 LET x=3+4*s
 260 LET y=0
 270 PRINT TAB x;
 280 FOR i=1 TO 5-s
 290 PRINT "#EXT SHI 2 GRA CD ##EXT SHI 0";
 300 NEXT i
 310 PRINT : PRINT TAB x;
 320 FOR i=1 TO 5-s
 330 PRINT "EXT SHI 4 GRA A EXT SH1 2 GRA EF EXT SHI 4 GRA
B# EXT SHI 0";
 185 NEXT i
 500 LET g=g+(g<27)*(INKEY$='c')-(g>1)*(INKEY$="x")
 510 PRINT AT v,g;"#EXT SH1 2 GRA QRS# EXT SH1 0";AT y,x;"#
###";AT y+1,x;"####"
 520 LET x=x+(y>3)-(y<4)+2*SGN (g-x)*(RND>.5): IF x>28 THEN
 LET x=0
 530 LET y=y+1
 540 PRINT AT y,x;"GRA G EXT SHI 1 GRA CD EXT SHI 0 GRA H G
RU";AT y+1,x;"GRA I EXT SHI 1 GRA EF EXT SHI 0 GRA J GRU"
 550 IF y=v-2 THEN  GO TO 1400
 560 IF RND>.15 THEN  GO TO 600
 570 FOR i=y+2 TO v-1: BEEP .02,v-i: PRINT AT i,x+1;"##";AT
 i+1,x+1;"GRA TU GRU": NEXT i
 580 PRINT AT i,x+1;"##"
 590 IF g=x OR g=x-1 THEN  GO TO 1300
 600 IF INKEY$<>"m" THEN  GO TO 650
 610 LET m=150: IF g=x OR g=x-1 THEN  LET m=157-8*v
```

```
 620 PLOT 19+8*g,8: DRAW 0,m: BEEP .1,0: IF m<150 THEN  GO
TO 1000
 630 DRAW  INVERSE 1;0,1: DRAW  INVERSE 1;0,-m-1
 640 REM tweede cyclus
 650 LET q=q+(q<27)*(INKEY$="c")-(q>1)*(INKEY$="z")
 660 PRINT AT v,g;"#EXT SHI 2 GRA ORS# EXT SHI 0";AT y,x;"#
###";AT y+1,x;"####"
 670 LET x=x+(y>3)-(y<4)+2*SGN (g-x)*(RND>.5): IF x<28 THEN
  LET x=0
 680 LET y=y+1
 690 PRINT AT y,x;"GRA F EXT SHI 1 GRA CD EXT SHI 0 GRA I G
RU";AT y+1,x;"GRA M EXT SHI 1 GRA EF EXT SHI 0 GRA N GRU"
 700 IF y=v-2 THEN  GO TO 1400
 710 IF RND>.15 THEN  GO TO 750
 720 FOR i=y+2 TO v-1: BEEP .01,v-i: PRINT AT i,x+1;"##";AT
 i+1,x+1;"GRA TU GRU": NEXT i
 730 PRINT AT i,x+1;"##"
 740 IF g=x OR g=x-1 THEN  GO TO 1300
 750 IF INKEY$<>"m" THEN  GO TO 500
 760 LET m=150: IF g=x OR g=x-1 THEN  LET m=157-8*y
 770 PLOT 19+8*g,8: DRAW 0,m: BEEP .1,0: IF m<150 THEN  GO
TO 1000
 780 DRAW  INVERSE 1;0,1: DRAW  INVERSE 1;0,-m-1
 790 GO TO 500
1000 FOR y=y+1 TO v-2
1005 BEEP .02,y/2
1010 PRINT AT y-1,x;"####";AT y,x;"EXT SHI 2 GRA O EXT SHI
 6 GRA CD EXT SHI 2 GRA F EXT SHI 0";AT y+1,x;"#EXT SHI 6 GR
A EF EXT SHI 0 #"
1015 NEXT y
1020 PRINT AT y-1,x;"####"
1200 LET s=s+1
1210 PRINT AT y,x;"####";AT y+1,x;"####";AT 0,0;: IF s<5 TH
EN  GO TO 250
1220 IF s=5 THEN  GO TO 1500
1300 PRINT AT v,g+1;"GRA ORS GRU"; OVER 1;AT v,x+1;"GRA TU
GRU": FOR i=1 TO 11: BEEP .1,-10: PRINT  OVER 1;AT v,g;"#GR
A ORS GRU": BEEP .02,4: NEXT i
1310 PAUSE 50
1320 CLS
1330 RUN 220
1400 PRINT AT y,x;"####";AT y+1,x;"####";AT 0,0;
1410 IF ABS (x-g)>1 THEN  GO TO 250
1420 PRINT AT v,g+1;"EXT SHI 2 GRA ORS EXT SHI 0"; OVER 1;A
T y+1,x;"GRA G EXT SHI 1 GRA CD EXT SHI 0 GRA H GRU";AT y+2
,x;"EXT SHI 1 GRA IEFJ GRU": FOR i=1 TO 11: BEEP .1,-10: PR
INT  OVER 1;AT v,g;"EXT SHI 2 #GRA ORS EXT SHI 0": BEEP .02
,4: NEXT i
1430 PAUSE 50
1440 RUN 100
1500 PRINT AT 5,0;"Gefeliciteerd#-#je#hebt#het#####ALIEN#ko
nvooi#totaal#vernietigd"
1510 BEEP .3,1: BEEP .05,0: BEEP .05,-1: BEEP .5,9: BEEP .0
5,0: BEEP .05,-1: BEEP .6,1: BEEP .5,10
1520 PRINT  FLASH 1; PAPER 2; INK 7;AT 10,0;"PAS#OP!#-#DAAR
#ZIJN#ZE#WEER..."
1530 FOR i=1 TO 3: BEEP 1,20: BEEP .1,0: NEXT i
1540 RUN 100
```

5.4 Spectrum-Invaders
(Copyright Beam Software)

Het is moeilijk te geloven, maar het populairste 'arcade'-spel aller tijden, het spel dat aan de wieg van alle videospelletjes heeft gestaan, kunt u hier op uw eigen Spectrum spelen. Deze versie van Spectrum Invaders biedt het volgende:

* 24 heen en weer bewegende invaders;
* vloeiende bewegingen op het scherm;
* schilden om u zich achter te verschuilen;
* gevaarlijke bommen van de vijand;
* laservuurkracht;
* score wordt continu bijgehouden;
* geluidseffecten;
* het spel wordt na elke golf moeilijker.

In het spel gebruikt u de 'Z'- en 'C'-toetsen om de laserbasis naar links en rechts te bewegen. Met de 'M'-toets vuurt u raketten af.
De verdediging van de planeet ligt in uw handen, aardbewoner!!

Opmerkingen
Een van de problemen bij het schrijven van een dergelijk spel in BASIC is het grote aantal benodigde variabelen, gecombineerd met de eis van snelle actie.
Het is eigenlijk niet mogelijk om dit soort BASIC-programma's te schrijven waarin 27 paar variabelen (de x, y coördinaten van 24 invaders, laserbasis, vijandelijke bommen en laserraketten) moeten worden bijgehouden om toch nog een redelijk 'arcade'-spel over te houden. Het zou dubbel zo lastig zijn om ook nog alles op het scherm te laten bewegen en dan ook nog bij te houden welke invader geraakt is bijvoorbeeld.
Dit verbazingwekkende programma is toch mogelijk dank zij twee eigenschappen van de Spectrum:

* een zeer lange 'string' op het scherm zetten kost nauwelijks meer tijd dan een korte 'string'. Het gaat zeker stukken sneller dan een aantal korte 'strings';
* met de unieke 'string-splitsingsmogelijkheden' van de Spectrum kan een gedeelte uit een string worden benoemd bijvoorbeeld:
  LET a$ = (2 to 6) = "HALLO"

De basis van dit programma wordt gevormd door twee strings, a$ en b$, die voortdurend gehanteerd worden. In a$ staan alle invaders opgeslagen, die kunnen dus in een keer op het scherm gezet worden enz. b$ is eigenlijk precies gelijk aan a$, hij heeft dezelfde lengte, bevat evenveel invaders enz., maar in b$ staan die invaders in een net iets andere positie opgeslagen. Door snel te wisselen tussen a$ en b$ lijkt het alsof de invaders hun armen, poten enz. bewegen.
Door a$ (en b$) op verschillende posities te laten beginnen kunt u in een keer alle invaders op het scherm krijgen en de oude uitvegen.

Als een invader door een laserstraal geraakt is dan wordt met behulp van de 'string splitsingstechniek' zoals die boven zijn omschreven, zijn plaats in de string opgevuld met spaties. Het einde van de string wordt voortdurend gecontroleerd om te kijken of er alleen maar spaties in voorkomen (dat zou namelijk betekenen dat alle invaders op de onderste rij geraakt zijn). is dat het geval dan wordt de splitsingstechniek gebruikt om de string in te korten tot de nieuwe maximale lengte.
Zo wordt niet alleen voorkomen dat er problemen ontstaan als een string onder de laatste regel van het scherm dreigt te komen, maar het geeft ook aan of een string een laserbasis geraakt heeft. Een klein bijkomend voordeel is dat naarmate de string korter wordt, minder tijd nodig is om het op het scherm te zetten, zodat het spel naar het einde toe steeds sneller gaat lopen.

Programma-opbouw

100 - 160 zet VPK's op

200 - 625 initialisatie
het belangrijkste is het opzetten van a$ en
b$ met behulp van de subroutine op 9000-9030

1000-1200 eerste gedeelte hoofdlus
$ op scherm, laserbasis aanpassen als nodig, laserstraal en positie vijandelijke bommen aanpassen

2000-2990 tweede deel hootdlus
identiek aan eerste deel maar voor b$ aanmaak nieuwe bom

3000-3530 invaders veranderen van richting
controleer of a$ gesplitst moet worden

5000-5220 effect van raak laserstraalschot

6000-6530 effect van rake vijandelijke bom

8500-8550 einde spel

Programma

```
  10 REM invaders
 100 DATA 0,31,63,63,123,113,225,225,0,248,252,252,222,142,
135,135,255,127,111,111,199,192,240,240,255,254,246,246,227
,3,15,15
 110 DATA 7,63,127,65,193,199,255,124,224,252,254,131,131,2
27,254,62,124,127,51,48,112,88,136,248,254,204,142,26,17,31
,0,0
 120 DATA 31,63,63,115,113,225,225,255,248,252,252,206,142,
```

```
135,135,255,127,111,103,103,96,48,60,60,254,246,230,230,6,1
2,60,60
 130 DATA 7,63,127,193,193,199,127,124,224,252,254,130,131,
227,255,62,127,51,113,88,136,248,0,0,62,254,204,12,14,26,17
,31
 140 DATA 24,24,24,24,60,126,231,231,7,31,63,60,112,112,248
,248,224,248,252,60,14,14,31,31
 150 DATA 18,26,14,14,7,3,1,0,36,44,56,56,112,96,192,128
 160 FOR i=USR "a" TO USR "a"+167: READ x: POKE i,x: NEXT i

 200 BORDER 7: PAPER 7: INK 0: CLS
 210 PRINT AT 0,0;"score";AT 0,22;"levens#"
 220 LET f=2
 230 LET sc=0
 250 LET q=f: LET f=f+1: IF f=11 THEN  STOP
 300 LET c$="##################################"
 310 LET s=144
 320 GO SUB 9000
 330 LET k$=a$+b$+c$
 340 GO SUB 9000
 350 LET l$=a$+b$+c$
 360 GO SUB 9000
 370 LET m$=a$+b$+c$
 380 GO SUB 9000
 390 LET n$=a$+b$+c$
 400 LET a$=k$+l$+l$+l$: LET a$=a$( TO 345)
 410 LET b$=m$+n$+m$+n$: LET b$=b$( TO 345)
 500 LET k$="GRA 4 SHI 4 37 SHI 7 GRU ###": LET l$="GRA SHI
 6 SHI 6 366 GRU ###": LET m$="GRA 72#1 SHI 4 GRU ###": LET
 n$=""
 510 PRINT  INK 4;AT 16,2;k$;k$;k$;k$;l$;l$;l$;m$;m$;m$;
m$
 590 LET l=0
 600 LET r=0: LET s=10: LET t=1
 610 LET u=0: LET v=0: LET x=0: LET y=0
 615 LET d=21: LET e=12
 620 PRINT AT 20,30;c$
 625 LET l1=3
1000 FOR p=r TO s STEP t
1010 BEEP .005,1: PRINT AT q,0;c$( TO p);a$;AT 20,1;"# GRA
0##################################R38# GRU"
1020 IF u<q THEN  GO TO 1050
1035 IF SCREEN$ (u,v)<>"#" THEN  GO SUB 5000
1040 IF u=q THEN  PRINT AT u,v;"GRA 0 GRU";AT u+1,v;"#"
1050 LET l=l+(l<28)*(INKEY$="e")-(l>0)*(INKEY$="z")
1060 IF u<>q-1 OR INKEY$<>"m" THEN  GO TO 1080
1070 LET u=20: LET v=l+1
1100 IF d=20 THEN  GO TO 1200
1110 IF SCREEN$ (d,e)<>"#" THEN  GO SUB 6000
1115 IF SCREEN$ (d,e+1)<>"#" THEN  GO SUB 6000
1120 IF d=20 THEN  PRINT AT d-1,e;"##"; INK 2;AT d,e;"GRA
1U GRU"
1200 LET u=u-1
2010 PRINT AT q,0;c$( TO p);b$;AT 20,1;"# GRA 0############
##################R38# GRU"
```

```
2020 IF u<q THEN  GO TO 2050
2035 IF SCREEN$ (u,v)<>"#" THEN  GO SUB 5000
2040 IF u>=q THEN  PRINT AT u,v;"GRA 0 GRU";AT u+1,v;"#"
2050 LET l=l+(l<28)*(INKEY$="c")-(l>0)*(INKEY$="z")
2060 IF u<q-1 OR INKEY$<>"m" THEN  GO TO 2080
2070 LET u=20: LET v=l+1
2100 IF d<=20 THEN  GO TO 2130
2110 LET w=INT (6*RND): LET x$=a$(LEN a$-4*w-2): IF x$<>"GR
A C GRU" AND x$<>"GRA G GRU" THEN  GO TO 2130
2120 LET d=q+1+INT (LEN a$/32): LET e=p-4*w+20+t
2125 IF e=31 THEN  LET e=30
2130 IF d=20 THEN  PRINT AT 20,e;"##"
2200 LET u=u-1
2210 LET d=d+1
2999 NEXT p
3000 IF t=-1 THEN  GO TO 3500
3010 LET t=-1: LET r=9: LET s=0
3020 GO TO 1000
3500 IF LEN a$>55 THEN  GO TO 3510
3505 IF a$(23 TO )=c$ THEN  GO TO 250
3510 IF a$(LEN a$-32 TO )=c$ THEN  LET a$=a$( TO LEN a$-96)
: LET b$=b$( TO LEN a$)
3520 IF 32*q+LEN a$>609 THEN  GO TO 8500
3525 LET r=0: LET s=10: LET t=1
3530 PRINT AT q,0;c$: LET q=q+1: GO TO 1000
5000 IF ATTR (u,v)=60 THEN  GO TO 5100
5005 IF ATTR (u,v)=58 THEN  GO TO 6000
5010 LET h=v-p+32*(u-q-1)
5015 BEEP .05,8
5020 IF h>=0 THEN  GO SUB 5200
5030 LET h=h+32: IF h<LEN a$ THEN  GO SUB 5200
5040 LET h=h+32: IF h<LEN a$ THEN  GO SUB 5200
5050 LET h=4-INT (h/96)
5060 LET sc=sc+100*h: PRINT AT 0,6;sc
5100 REM explosie
5110 PRINT  INK 6;AT u,v;"GRA BA GRU";AT u+1,v;"GRA SR GRU"
: BEEP .05,10: PRINT AT u,v;"##";AT u+1,v;"##"
5120 LET u=0
5130 RETURN
5200 IF h=0 THEN  LET h=1
5205 LET a$(h TO h+2)="###"
5210 LET b$(h TO h+2)="###"
5220 RETURN
6000 REM test wat alien kogel geraakt heeft
6005 IF d=20 THEN  GO TO 6018
6010 PRINT AT d-1,e;"##"; INK 3;AT d,e;"GRA AB GRU": BEEP .
05,-10: PRINT AT d,e;"##"
6012 IF (v=e OR v=e+1) AND d=u THEN  PRINT AT u,v;"#";AT u+
1,v;"#": LET u=0
6015 LET d=21: RETURN
6020 PRINT AT d-1,e;"##"; INK 2;AT d,e;"GRA AB GRU": BEEP .
05,10: PRINT AT d,e;"##"
6030 FOR k=1 TO 10: PRINT  OVER 1;AT 21,l;"GRA R35 GRU": BE
EP .1,0: NEXT k
6040 LET l=l-1
6050 PRINT AT 0,29;"###";AT 0,29;"( TO l)
```

```
6060 IF l1=0 THEN  GO TO 8510
6500 REM projectielen raken elkaar
6510 IF d<=20 THEN  PRINT  INK 6;AT d,e;"GRA BA GRU";AT d+1
,e;"GRA SR GRU": BEEP .05,10: PRINT AT d,e;"##";AT d+1,e;"#
#"
6520 LET u=0: LET d=21
6530 RETURN
8500 PRINT AT q,0;c$;a$
8510 PRINT AT 5,0;"##De#invaders#landen#op#aarde.": PRINT '
##Alles#is#verloren!!"
8520 INPUT "Wil#je#nog#een#keer?";n$
8530 IF LEN n$=0 THEN  GO TO 8520
8540 IF n$(1)="j" OR n$(1)="J" THEN  RUN 200
8550 STOP
9000 LET a$=CHR$ (s)+CHR$ (s+1)+"##": LET a$=a$+a$+a$+a$+a$
+a$+"########"
9010 LET b$=CHR$ (s+2)+CHR$ (s+3)+"##": LET b$=b$+b$+b$+b$+
b$+b$+"#######"
9020 LET s=s+4
9030 RETURN
```

5.5 Meteorenstorm
(Copyright Beam Software)

'Meteorenstorm' zet u achter de stuurknuppel van een ruimteschip dat door de asteroidengordel raast. Op het scherm heeft u een fraai uitzicht op het zonnestelsel vanuit het midden van de ruimte.
In de verte kunt u Saturnus met haar ringen zien, en daar achter de talrijke sterren van de nabij gelegen melkwegstelsels.
Plotseling waarschuwt de boordcomputer voor een Meteorenstorm! Voor uw ogen ziet u een meteoor steeds groter worden terwijl u er recht op af vliegt. De alarmerende situatie wordt bevestigd door de boordradar die de snel naderende meteoor op het scherm heeft.
'Meteorenstorm' is een 3D-computerspel op zijn best.

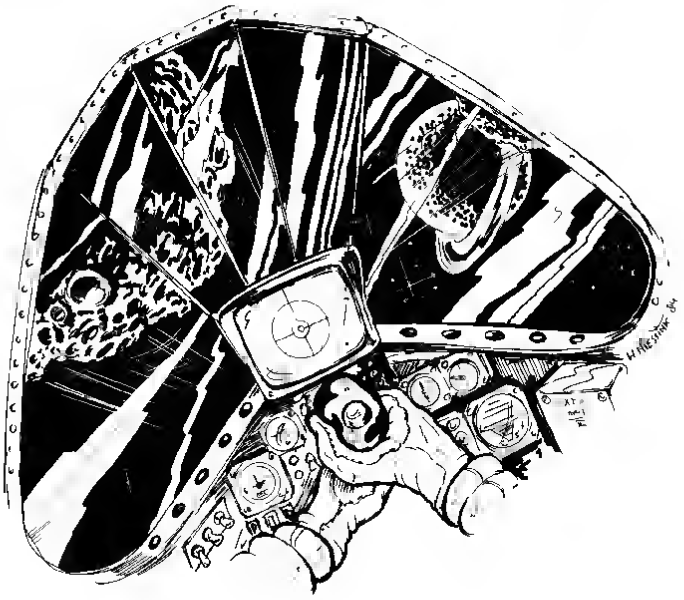
Gebruik van het programma
Nadat RUN + ENTER ingetoetst zijn komt het bekende instrumentenpaneel tot leven. Een richtkruis in het midden van het scherm markeert het doel waarop de laserstraal gericht staat. Met de volgende toetsen kunt u de straal besturen:

'I' en 'P'	naar links en rechts
'W' en 'X'	omhoog en omlaag
'O'	vuur

Onthoud wel dat net als in de meeste BASIC-programma's, maar één toets tegelijk kan worden gebruikt.

Het richtkruis loopt door over het scherm - dat wil zeggen dat als u te ver naar rechts gaat op het scherm, het kruis automatisch weer links terugkomt.

De laser kan altijd afgevuurd worden, maar als er geen asteroiden geraakt worden dan verdwijnt de straal met een dol geluid in de ruimte. Raakt u wel iets, dan kaatst de straal zonder gevaar terug.
Naarmate u verder in de meteorenstorm verzeild raakt, komen de asteroiden steeds sneller op u af. Hoeveel kunt u er afschieten als u maar drie levens heeft?



Programma-aanwijzingen
Dit programma maakt uitgebreid gebruik van de OVER-opdracht van de Spectrum en demonstreert de mogelijkheid om VPK's opnieuw te gebruiken.
In het begin van het programma worden de VPK's gebruikt om het instrumentenpaneel, het radarscherm en Saturnus te tekenen. Dit is de enige keer in het hele programma dat ze op het scherm gezet worden. Hierna worden de VPK's onmiddellijk gebruikt om de asteroiden op het scherm te zetten!
Dit is mogelijk omdat het scherm niet afhankelijk is van karakters - als iets eenmaal in het schermgeheugen staat dan blijft het scherm dit tonen totdat deze informatie door andere vervangen wordt.

Het grote voordeel van het gebruik van OVER moet nu duidelijk zijn. Door heel goed bij te houden wat er op het scherm is gezet en door er voor te zorgen dat iets alleen uitgeveegd wordt als daarna op dezelfde positie de originele informatie weer terug gezet wordt, kunt u alle informatie in het oorspronkelijke formaat bewaren.

(Als u iets twee keer met een OVER-instructie op het scherm zet krijgt u weer te zien wat oorspronkelijk op die plaats op het scherm stond - probeer maar!)

Programma-opbouw

100 - 200	zet volgende karakters op ronde hoeken ruimteschip op radarscherm Saturnus kleine meteoren I (ABCD) kleine meteoren II (EFG) meteoren III (HIJK) meteoren IV (LM) explosie I (NOP) explosie II (QRS)
200 - 340	zet instrumentenpaneel op
400 - 590	zet nieuwe karakters op initialiseer en FLITS-melding b$ opzetten kost de meeste tijd, deze array bevat willekeurige meteoorvorm die gebruikt wordt bij laatste botsingexplosie belangrijke variabelen: x,y richtkruispositie ox,oy oude richtkruispositie li aantal levens ms meteoorstand (0-9) di moeilijkheidsgraad mx,my meteoorpositie v INK-kleur onderrichtkruis
900 - 1100	hoofdlus: test toetsenbord, beweeg richtkruis
1200 - 1500	laservuur
3000 - 3100	subroutine om radar aan te passen
4000 - 4880	subroutines om meteoor in verschillende afmetingen op scherm te zetten
5000 - einde	routine voor botsing met meteoor

Opmerkingen
Door het veelvuldig gebruik van DATA-statements wordt veel geheugenruimte in beslag

genomen. Als u geen speciale maatregelen genomen zou hebben, dan zou dit programma niet in 16K passen.
Elk getal in een DATA-statement, maar ook in de rest van het BASIC-programma, bestaat niet alleen uit de karakters die het getal vormen, maar wordt ook nog gevolgd door een rij van 6 bytes die de Spectrum-ROM nodig heeft om de waarde van dat getal te kunnen bepalen. Een DATA-statement met 32 getallen van gemiddeld 3 cijfers zal dus ongeveer 300 geheugenplaatsen in beslag nemen. Als u nu vaak voorkomende getallen vervangt door variabelen dan bespaart u een aanzienlijke hoeveelheid geheugenruimte.
In het begin van dit programma worden dan ook de getallen 0 en 1 vervangen door variabelen.

Programma

```
  10 REM meteoor
  90 LET z=0: LET u=1
 100 DATA 255,248,224,192,192,128,128,128,255,31,7,3,3,u,u,
u,128,128,128,192,192,224,248,255,u,u,u,3,3,7,31,255
 110 DATA z,z,128,192,231,255,255,255,z,z,z,z,192,240,60,25
5,u,3,7,14,z,z,z,z,240,224,128,z,z,z,z,z
 120 DATA z,3,7,15,15,31,31,127,z,192,240,248,248,254,253,2
49,127,220,192,115,31,7,3,z,195,4,60,248,248,240,192,z
 130 DATA z,z,u,3,7,7,7,z,z,4,6,15,31,31,63,14,z,4,14,31,63
,63,63,31,30,30,28,12,z,z,z,z,z
 140 DATA 31,63,31,31,15,6,z,z,z,z,z,z,128,192,224,224,224,
224,224,192,z,z,z,z
 150 DATA 3,23,55,63,127,127,127,127,128,224,240,240,224,24
8,248,252,127,63,63,127,63,31,11,z,252,252,248,224,240,224,
128,z
 160 DATA 63,127,255,127,255,255,127,63,252,248,252,254,254
,248,248,252
 170 DATA 195,153,60,102,102,60,153,195,195,99,39,60,60,180
,231,231,128,60,230,255,91,200,221,119
 180 DATA 38,62,246,254,108,71,227,62,115,55,246,180,28,188
,246,99,221,118,103,238,184,115,255,128
 200 BORDER 5: INK 5: PAPER 6: CLS
 210 RESTORE 100
 220 FOR i=USR "a" TO USR "m"-u
 230 READ x: POKE i,x: NEXT i
 240 DIM s(10)
 300 PRINT AT z,z;"GRA A GRU";AT z,31;"GRA B GRU";AT 18,z;"
GRA C GRU";AT 18,31;"GRA D GRU"
 310 PRINT "GRA SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 S
HI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8
 SHI 8 SHI 8 SHI 8 SHI 8 SHI 8 GRU": PRINT "GRA SHI 8 A####
############B SHI 8 SHI 8 SHI 8 SHI 8 A########B SHI 8 SHI 8
C################D SHI 8 SHI 8 SHI 8 SHI 8 C########D SHI 8 G
RU"
 320 PRINT  INK z;AT 20,2;"GRA EF GRU";AT 21,2;"GRA GH GRU"
;AT 21,23;"Ievens#3"
 330 FOR i=u TO 20: PRINT  INK 2;AT u+16*RND,u+29*RND;".":
NEXT i
```

```
 340 PRINT  INK 3;AT 6,10;"GRA IJ GRU";AT 7,10;"GRA KL GRU"

 400 FOR i=USR "a" TO USR "t"-u
 410 READ x: POKE i,x: NEXT i
 420 PRINT  OVER u; FLASH u; INK 2;AT 10,4;"KLAARMAKEN#VOOR
#VLUCHT#IN";AT 11,10;"METEOREN#STORM"
 430 LET b$="": FOR j=u TO 17
 435 BEEP .05,z
 440 LET a$="#"
 450 FOR i=u TO 15
 460 IF RND>.6 THEN  LET a$=a$+CHR$ (144+12*RND)
 470 LET a$=a$+"##"
 475 BEEP .005,10
 480 NEXT i
 490 LET b$=b$+a$( TO 29)+"#": NEXT j
 500 PRINT  OVER u; FLASH z;AT 10,4;"KLAARMAKEN#VOOR#VLUCHT
#IN";AT 11,10;"METEOREN#STORM"
 520 LET ox=9: LET oy=15: LET ov=7
 530 LET x=9: LET y=15: LET v=7
 540 INK z: PRINT AT x,y;"+"
 550 LET i1=3: LET di=u: LET nm=z
 560 LET ms=z: LET nm=nm+u
 580 LET mx=u+RND*17: LET my=u+RND*29
 590 LET mi=u
 900 LET a$=INKEY$
 910 LET ms=ms+di/7+nm/50
 920 IF INT ms=9 THEN  GO TO 5000
 930 GO SUB 3000
 940 GO SUB 4000+100*INT ms
 950 LET w=ATTR (x,y)-8*INT (ATTR (x,y)/8): IF w<>0 THEN  L
ET v=w
 960 IF a$="" OR a$="o" THEN  GO TO 1070
1000 LET y=oy+(a$="p")-(a$="i")
1010 IF y<u THEN  LET y=30
1020 IF y>30 THEN  LET y=u
1030 LET x=ox+(a$="s")-(a$="w")
1040 IF x<u THEN  LET x=17
1050 IF x>17 THEN  LET x=u
1060 LET v=ATTR (x,y)-8*INT (ATTR (x,y)/8)
1070 PRINT  OVER u; INK ov;AT ox,oy;"+"; INK z;AT x,y;"+"
1090 LET ox=x: LET oy=y: LET ov=v
1100 IF a$<>"o" THEN  GO TO 900
1200 FOR i=u TO 6: BEEP i*i/100,10/i: NEXT i
1250 IF v<>u THEN  GO TO 900
1260 LET x1=x-u: IF x1<z THEN  LET x1=z
1270 LET y1=y-u: IF y1<z THEN  LET y1=z
1275 LET x2=x+u: IF x2>18 THEN  LET x2=18
1280 OVER u: FOR i=u TO 10: BEEP .03,3: PRINT AT x,y;"GRA N
 GRU": BEEP .01,u: PRINT AT x1,y1;"GRA OFO GRU";AT x,y1;"GR
A ONO GRU";AT x2,y1;"GRA RGR GRU": NEXT i: OVER z
1290 DIM s(10): LET mi=2
1300 FOR i=INT ms TO z STEP -u
1310 LET ms=i: GO SUB 4000+100*i: GO SUB 3000
1320 NEXT i
1340 DIM s(10)
```

```
1360 LET ov=z: LET v=z
1370 PRINT  INK z; OVER z;AT 20,23;nm+11-3;AT 20,26;"raak"
1380 PRINT  OVER u; INK 3;AT 6,10;"##";AT 7,10;"##"
1500 GO TO 560
3000 IF INT ms<4 THEN  PRINT  INK u;AT 20,14-ms;"# GRA B GR
U #"
3010 IF INT ms=4 THEN  PRINT  INK u;AT 20,9;"## GRA B GRU #
#"
3020 IF INT ms>4 AND INT ms<8 THEN  PRINT  INK u;AT 20,18-2
*INT ms;"## GRA B GRU ##"
3030 IF INT ms=8 THEN  PRINT  INK u;AT 20,4;"GRA B GRU ##"
3040 RETURN
4000 IF s(1)=u THEN  RETURN
4010 PRINT  OVER u; INK m1;AT mx,my;"GRA A GRU"
4020 LET s(1)=u
4030 RETURN
4100 IF s(2)=u THEN  RETURN
4110 PRINT  OVER u; INK m1;AT mx,my;"GRA B GRU"
4120 LET s(2)=u
4130 RETURN
4200 IF s(3)=u THEN  RETURN
4210 PRINT  OVER u; INK m1;AT mx,my;"GRA BF GRU"
4220 LET s(3)=u
4230 RETURN
4300 IF s(4)=u THEN  RETURN
4310 LET mx4=mx+u
4320 PRINT  OVER u; INK m1;AT mx,my;"GRA BF GRU";AT mx4,my;
"GRA D GRU"
4330 LET s(4)=u
4340 RETURN
4400 IF s(5)=u THEN  RETURN
4410 PRINT  OVER u; INK m1;AT mx,my;"GRA C GRU";AT mx4,my;"
GRA E GRU"
4420 LET s(5)=u
4430 RETURN
4500 IF s(6)=u THEN  RETURN
4510 PRINT  OVER u; INK m1;AT mx,my;"GRA CF GRU";AT mx4,my;
"GRA EG GRU"
4520 LET s(6)=u
4530 RETURN
4600 IF s(7)=u THEN  RETURN
4610 PRINT  OVER u; INK m1;AT mx,my;"GRA HI GRU";AT mx4,my;
"GRA JK GRU"
4620 LET s(7)=u
4630 RETURN
4700 IF s(8)=u THEN  RETURN
4710 LET mx2=mx-u
4720 LET my6=my+2: IF my6>31 THEN  LET my6=31
4730 LET mx5=mx+2: IF mx5>18 THEN  LET mx5=18
4740 LET my3=my-u
4750 PRINT  OVER u; INK m1;AT mx2,my;"GRA CF GRU";AT mx,my3
;"GRA C SHI 8 SHI 8 GRU";AT mx,my6;"GRA F GRU";AT mx4,my3;"
GRA E SHI 8 SHI 8 GRU";AT mx4,my6;"GRA G GRU";AT mx5,my;"GR
A EG GRU"
4760 LET s(8)=u
```

```
4770 RETURN
4800 IF s(9)=u THEN  RETURN
4810 LET mx1=mx-2: IF my1<z THEN  LET my1=z
4820 LET my2=my-2: IF my<z THEN  LET my=z
4830 LET my7=my+3: IF my7>31 THEN  LET my7=31
4840 LET mx6=mx+3: IF mx6>18 THEN  LET mx6=18
4850 LET mx6=mx+3: IF mx6>10 THEN  LET mx6=18
4860 PRINT  OVER u; INK m; AT mx1,my;"GRA CF GRU";AT mx2,my
3;"GRA H SHI 8 SHI 8 GRU";AT mx2,my6;"GRA I GRU";AT mx,my2;
"GRA C GRU";AT mx,my3;"GRA SHI 8 SHI 8 SHI 8 GRU";AT mx,my6
;"GRA SHI 8 GRU";AT mx,my7;"GRA F GRU";AT mx4,my2;"GRA E GR
U";AT mx4,my3;"GRA SHI 8 SHI 8 SHI 8 GRU";AT mx4,my6;"GRA S
HI 8 GRU";AT mx4,my7;"GRA G GRU";AT mx5,my3;"GRA J SHI 8 SH
I 8 GRU";AT mx5,my6;"GRA K GRU";AT mx6,my;"GRA EG GRU"
4870 LET s(9)=u
4880 RETURN
5000 FOR i=z TO 16
5010 PRINT  OVER u; FLASH u;AT i+u,u;b$(30*i+u TO 30*i+30)
5020 NEXT i
5030 PRINT  FLASH u; OVER u;AT 20,1;"##################";AT
21,1;"##################"
5400 DIM s(10)
5500 FOR i=u TO 40: BEEP RND/10,10*RND-5
5510 IF i>31 THEN  GO SUB 8000-100*i
5520 NEXT i
5530 LET ov=4: LET v=4
6000 FOR i=z TO 16
6010 PRINT  OVER u; INK 2; FLASH z;AT i+1,1;b$(30*i+1 TO 30
*i+30)
6020 NEXT i
6030 PRINT  FLASH z; INK 5; OVER 1;AT 20,1;"# EXT SHI O ###
############ EXT SHI 5 #";AT 21,1;"#EXT SHI O###############
# EXT SHI 5 # EXT SHI O"
6035 PRINT  OVER u;AT 20,4;"GRA B GRU"
6040 LET li=li-u
6050 PRINT  OVER z; INK z;AT 21,30;li
6060 IF li=z THEN  STOP
6070 DIM s(10)
6200 GO TO 560
```

5.6 Kidnap

(Copyright Beam Software)

Het is uw taak om de levende wezens op het planetoide-oppervlak te beschermen. Het is aan u om de vijandelijke invasie tegen te houden.

U moet ten koste van alles zien te voorkomen dat de wezens die onder uw hoede staan gekidnapt worden door de vijand. Hun overlevingskansen en het voortbestaan van de planeet zijn in uw handen.

De planetoide onder uw bewaking is niet zo groot. U kunt er gemakkelijk omheen vliegen, maar u kunt slechts in één richting reizen.

De vijandige kidnappers hebben een warmtezoekende raket op u afgevuurd om de aan-

dacht af te leiden van hun boze plannen. Lukt het u de raket te ontwijken, de vijand neer te schieten en de wezens te beschermen?

Gebruik van het programma

U bestuurt het ruimteschip in het midden van het scherm. De krachten waarmee u vooruit snelt maken elke zijwaartse beweging onmogelijk.
U kunt dus alleen maar omhoog (de 'W'-toets) en omlaag (de 'X'-toets).
U heeft een laserkanon tot uw beschikking (de 'P'-toets) waarmee u de raketten en de vijand zelf kunt uitschakelen. Maar de rust is van korte duur, zo gauw u een kidnapper neerschiet verschijnt er een nieuwe.
Uw ruimteschip kan wat beschadigingen verwerken, maar na drie treffers is het niet meer operationeel.

Programma-aanwijzingen

In BASIC heeft u geen instructies om de bewegingen van het ruimteschip tegen de oppervlakte van de planeet na te bootsen. En zelfs als dergelijke instructies bestonden (bijv. SCREEN$, die u hier niet kunt gebruiken omdat deze instructie een vrij programmeerbaar karakter niet kan uitlezen) dan zouden die nog te traag zijn om het hele scherm te laten schuiven.
Dit probleem kunt u alleen maar met een routine in machinetaal oplossen. In het DATA-statement op regel 3000 staat de machinetaal om het scherm in horizontale richting te kunnen schuiven (36 bytes lang).

Deze routine schuift elke kolom op het scherm een plaats door, het karakter op positie 0,1 komt op 0,0, die op 0,0 gaat naar 0,31 enz.
Daarna doet deze routine hetzelfde met de kleurattributen zodat alles zijn kleur houdt.
Deze routine is niet plaatsgebonden. U heeft dus geen kennis van machinetaal nodig om hem met behulp van een soortgelijke programmering als regel 140 en 150 overal in het geheugen te zetten waar u hem wilt hebben.
Het kan interessant zijn om deze routine ook in andere programma's te gebruiken.

De rest van het programma houdt zich voornamelijk bezig met het aanpassen van de variabelen die betrekking hebben op de posities van het ruimteschip (sx,sy), van de aanvallers (ax,ay) en van de bommen (bx,by). Door het voortdurend heen en weer geschuif is dit een tijdrovende, maar essentiele taak.

Programma-opbouw

100 - 150 zet variabelen en VPK's op

160 - 360 teken landschap en plaats mannetjes in toren

1000-1020 teken schip, aanvaller en bom

5000-5360 hoofdlus
test schipbeweging
zorg dat aanvaller wezen zoekt
richt bom op schip
vuur laser af als toets ingedrukt is
test eventuele botsingen

7000-7330 kidnaplus
test overlevenden enz

8500-8760 beschadiging schip routine
inclusief einde programma

Programma

```
  10 REM kidnap
 100 RESTORE 2000
 110 FOR i=USR "a" TO USR "k"-1
 120 READ x: POKE i,x
 130 NEXT i
 140 RESTORE 3000: CLEAR 32500
 150 FOR i=32500 TO 32535: READ x: POKE i,x: NEXT i
 160 BORDER 1: PAPER 1: INK 0: CLS
 170 PLOT 0,8
 180 LET y=8
 190 LET p=-1
 200 FOR j=1 TO 3
 210 LET a=3+INT (7*RND)
 220 GO SUB 8100
 230 DRAW 4,15-y: DRAW 8,0
 240 LET y=15
 250 LET p=p+a+1
 260 FOR f=1 TO INT (4*RND)+1: PRINT  INK 5; OVER 1;AT 20-f
,p;"GRA C GRU": NEXT f: PRINT  INK 3;AT 20-f,p;"GRA B GRU";
AT 19-f,p;"GRA A GRU"
 270 NEXT j
 280 LET a=30-p
 290 GO SUB 8100
 300 DRAW 11,8-y
 310 LET sx=3: LET sy=10
 320 LET ax=INT (15*RND): LET ay=26
 330 LET bx=INT (RND*15): LET by=30: IF bx=ax THEN  GO TO 3
30
 340 LET t=0
 350 LET e=0: LET h=3
 360 LET dm=0: LET sc=0
1000 PRINT  INK 1;AT sx,sy;"GRA #FGHI GRU"
1010 PRINT  INK 4;AT ax,ay;"GRA #DE GRU"
1020 PRINT  INK 2;AT bx,by;"GRA #J GRU"
2000  DATA 1,57,189,187,146,254,124,124,56,56,56,104,76,68,
68,198: REM vorm wezens
```

```
2010 DATA 126,36,60,24,60,102,195,255: REM vorm platform
2020 DATA 63,99,193,193,127,28,7,3,252,198,131,131,254,56,2
24,192: REM vorm aliens
2030 DATA 255,63,15,7,3,7,255,0,192,240,252,255,255,255,240
,3,0,0,31,249,224,255,252,240,0,0,0,192,248,255,0,0: REM vo
rm schip
2040 DATA 28,28,244,199,227,47,56,56
3000 DATA 33,1,64,17,0,64,26,1,31,0,237,176,18,35,19,62,88,
188,32,242,6,22,197,26,1,31,0,237,176,18,35,19,193,16,243,2
01
5000 LET v=USR 32500
5010 LET x$=INKEY$
5020 LET nx=sx+(x$="p")*(sx<18)-(x$="w")*(sx>0)
5030 IF nx=sx THEN  GO TO 5060
5040 LET v$=SCREEN$ (nx,sy+1): LET w$=SCREEN$ (nx,sy+2): LE
T v$=v$+SCREEN$ (nx,sy+3): LET v$=v$+w$+x$
5050 IF v$(2 TO )="###" THEN  GO TO 8500
5060 IF SCREEN$ (nx,sy+4)<>"#" THEN  GO TO 8500
5070 PRINT  OVER 1;AT sx,sy-1;"GRA #FGHI GRU";AT nx,sy;"EXT
 SHI 6 GRA #FGHI EXT SHI 0"
5080 LET sx=nx
5090 LET by=by-1: IF by<0 THEN  LET by=31
5100 IF RND>.7 OR by=31 THEN  GO TO 5140
5110 LET nx=bx+(sx-1>bx)-(sx-1<bx)+(by-sy+2)
5120 PRINT  OVER 1;AT bx,by;"GRA #I GRU";AT nx,by;"EXT SHI
2 GRA #J EXT SHI 0"
5130 LET bx=nx
5140 IF bx=sx AND ABS (by-sy-1)<>1 THEN  GO TO 8500
5200 IF x$<>"p" THEN  GO TO 5290
5210 BEEP .1,4
5220 PLOT  INK 3; OVER 1;8*sy+43,170-8*sx
5230 DRAW  INK 3; OVER 1;212-8*sy,0
5240 BEEP .1,8
5250 PLOT  INK 0; OVER 1;8*sy+43,170-8*sx
5260 DRAW  INK 0; OVER 1;212-8*sy,0
5270 IF ax=sx AND ay>sy+4 AND ay<sy+30 THEN  FOR k=1 TO 11: BE
EP .1,5: PRINT  OVER 1;AT ax,ay-1;"GRA #DE GRU": NEXT k: LE
T ax=INT (15*RND): LET ay=29: PRINT AT ax,ay-1;"EXT SHI 4 G
RA #DE EXT SHI 0": LET sc=sc+100
5280 IF bx=sx AND by>sy+4 AND by<sy+30 THEN  FOR k=1 TO 5: BEE
P .1,10: PRINT  OVER 1;AT bx,by;"EXT SHI 2 GRA #J EXT SHI 0
": NEXT k: LET bx=1+INT (RND*14): LET by=30: PRINT AT bx,by
;"EXT SHI 2 GRA #J EXT SHI 0": LET sc=sc+20
5290 LET ay=ay-1: IF ay<0 THEN  LET ay=31
5300 IF RND>.7 OR ay>=29 THEN  GO TO 5000
5310 LET nx=ax+(ax<18)-(ax>18): IF ax=18 THEN  GO TO 5330
5320 LET v$=SCREEN$ (nx,ay+1): LET w$=SCREEN$ (nx,ay+2): IF
 v$+w$<>"##" THEN  GO TO 7000
5330 LET ny=ay+(ay=18)
5340 IF ny<>ay AND SCREEN$ (nx,ny+2)<>"#" THEN  GO TO 7000
5350 PRINT  OVER 1;AT ax,ay;"GRA #DE GRU";AT nx,ny;"EXT SHI
 4 GRA #DE EXT SHI 0"
5360 LET ay=ny: LET ax=nx
6000 GO TO 5000
7000 IF ABS (bx-ax)<=1 AND ABS (by-ay)<=1 THEN  FOR k=1 TO
```

```
11: BEEP .1,5: PRINT  OVER 1;AT ax,ay;"GRA #DE GRU": NEXT k
: LET ax=INT (15*RND): LET ay=29: PRINT AT ax,ay;"EXT SHI 4
 GRA #DE EXT SHI 0"
7010 IF ABS (sx-ax)<=2 AND ABS (by-sy)<=2 THEN  GO TO 8500
7020 PRINT  OVER 1;AT ax,ay;"GRA #DE GRU";AT ax-1,ay;"EXT S
HI 4 GRA #DE EXT SHI 0"
7030 LET ax=ax-1: LET ay=ay-1: LET by=by-1
7040 IF ay<0 THEN  LET ay=31
7050 IF by<0 THEN  LET by=31
7060 LET v=USR 32500
7070 PRINT  OVER 1;AT sx,sy-1;"GRA #FGH1 GRU";AT sx,sy;"EXT
 SHI 6 GRA #FGH1 EXT SHI 0"
7080 LET f=0
7090 FOR x=ax-3 TO ax+3
7095 IF x>20 THEN  GO TO 7125
7100 FOR y=ay+1 TO ay+3
7105 IF y<0 OR y>31 THEN  GO TO 7120
7110 IF ATTR (x,y)=11 THEN  PRINT AT x,y; INK 1;"#": LET f=
f+1
7120 NEXT y
7125 NEXT x
7130 IF f=2 THEN  GO TO 7200
7140 PRINT AT ax,ay+3; INK 1;"#";AT ax+1,ay+2;"###";AT ax+2
,ay+2;"###"
7145 LET ay=ay+1: PRINT  OVER 1;AT ax,ay-1;"GRA #DE GRU";AT
 ax,ay;"EXT SHI 4 GRA #DE EXT SHI 0"
7150 LET e=e+1: IF e=3 THEN  GO TO 8550
7160 GO TO 5000
7200 PRINT  OVER 1;AT ax,ay;"GRA #DE GRU";AT ax-1,ay;"EXT S
HI 4 GRA #DE EXT SHI 0"
7210 LET ax=ax-1: REM  LET ay=ay-1: IF ay<0 THEN  LET ay=31

7220 LET by=by-1: IF by<0 THEN  LET by=31
7230 LET v=USR 32500
7240 PRINT  INK 4;AT ax+1,ay;"GRA #A GRU";AT ax+2,ay;"GRA #
B GRU"
7250 FOR k=ax-1 TO 0 STEP -1
7260 PRINT  OVER 1; INK 4;AT k+1,ay;"GRA DE GRU";AT k+2,ay;
"GRA #A GRU";AT k+3,ay;"GRA #B GRU";AT k,ay;"GRA DE GRU";AT
 k+1,ay;"GRA #A GRU";AT k+2,ay;"GRA #B GRU"
7270 BEEP .2,15-k/2
7280 NEXT k
7290 LET e=0: LET h=h-1: IF h=0 THEN  GO TO 8550
7300 PRINT  OVER 1;AT sx,sy-1;"GRA #FGH1 GRU";AT sx,sy;"EXT
 SHI 6 GRA #FGH1 EXT SHI 0"
7310 PRINT  INK 1;AT 1,ay;"##";AT 2,ay;"##"
7320 LET ax=0: LET ay=ay-1: IF ay<0 THEN  LET ay=31
7330 GO TO 5000
8100 FOR i=0 TO 2*a-2
8110 LET ly=INT (36*RND)-18
8120 IF y+ly<0 THEN  LET ly=-y
8130 IF y+ly>22 THEN  LET ly=22-y
8140 DRAW 4,ly
8150 LET y=y+ly
8160 NEXT i
```

```
8170 RETURN
8500 PRINT  INK 5; FLASH 1;AT 21,0;"Schip#beschadigd"
8510 LET dm=dm+1: IF dm<3 THEN  GO TO 8700
8520 PRINT AT 21,16; FLASH 1; INK 7;"#ONHERSTELBAAR": FOR s
=22944 TO 23104
8530 IF PEEK s=11 THEN  BEEP .5,10: POKE s,9: POKE s+32,9
8540 NEXT s
8550 FOR k=1 TO 15
8560 PRINT  OVER 1;AT sx,sy-1;"# EXT SHI 6 GRA FGHI EXT SHI
 0": BEEP k/50,12-2*k: NEXT k
8570 PRINT AT 1,0; INK 7;"Wezens#niet#langer#bescher md.###J
e#hebt#gefaald#in#je#opdracht."
8580 PRINT  INK 2;AT 10,0;"Je#score=#";sc
8600 INPUT "Zou#je#het#nog#een#keer#willen##proberen?#";a$
8610 LET a$=a$+"#": IF a$(1)="j" THEN  RUN 100
8650 STOP
8700 FOR k=1 TO 15
8710 PRINT  OVER 1;AT sx,sy-1;"# EXT SHI 6 GRA FGHI EXT SHI
 0": BEEP .1,15-2*k: NEXT k
8720 PRINT  INK 1; FLASH 0;AT 21,0;"################"
8730 PRINT  OVER 1;AT ax,ay-1;"GRA #DE GRU";AT bx,by-1;"GRA
 #J GRU"
8740 LET ax=INT (15*RND): LET ay=26
8750 LET bx=INT (15*RND): LET by=30: IF bx=ax THEN  GO TO 8
750
8760 GO TO 1000
```

5 7 Snelwegkikker
(Copyright Beam Software)

U bent een arme kleine kikker die wanhopig probeert de snelweg over te steken op weg naar huis. Op de snelweg moet u vrachtwagens, motorfietsen en auto's zien te ontwijken, maar het ergste zijn de politiewagens die met hoge snelheid elk moment voorbij kunnen komen. Om de kikker te besturen gebruikt u de volgende toetsen:

'Q' naar boven
'A' naar beneden
'O' naar links
'P' naar rechts

U krijgt punten als u naar boven en naar beneden gaat, dus als u een hoge score wilt hebben dan kunt u de snelweg net zo vaak oversteken als u wilt. Als een kikker boven in het scherm aankomt dan is hij thuis, en dan probeert u het weer met de volgende kikker.

Hoe het programma werkt
Dit programma maakt uitgebreid gebruik van de vrij programmeerbare karakterset van de SPECTRUM. Als u daar te weinig vanaf weet dan zou u hoofdstuk 14 van de gebruiksaanwijzing kunnen lezen. Dit programma gebruikt 58 programmeerbare karakters, hoewel er

maar 21 mogelijk zijn. Om het de computer mogelijk te maken zoveel karakters te gebruiken moet het VPK-gebied heen en weer geschoven worden over de twee of meer karaktersets.
Hiervoor moet u eerst ruimte maken voor het aantal karakters dat u nodig heeft. Bij een 16K-machine met 21 vrij programmeerbare karakters bevindt het VPK-gebied zich op geheugenadres 32600 tot 32767. Dit zijn 168 bytes, of wel 8 bytes per karakter. Daar u 58 karakters nodig heeft voor uw spelletje moet u 464 bytes vrijmaken voor het VPK-gebied. Dit komt overeen met adres 32304 tot 32767, de eerste opdracht in uw programma wordt dus:

CLEAR 32303

N.B. alle gedefinieerde variabelen uit LET en DIM opdrachten gaan verloren na een CLEAR.

In dit programma heeft u drie grafische karaktersets:
De eerste twee bevatten elk 21 karakters, de derde slechts 16. Om nu een set te kunnen gebruiken moet u de systeem variabele VPK (adres 23675 en 23676) naar het juiste gebied laten wijzen.

32600 de eerste set (normale set)
32432 de tweede set (32600 - 168)
32304 de derde set (32432 - 128)

N.B. de derde set loopt gedeeltelijk over in de tweede omdat hij maar 16 karakters lang is.

Het kost wat rekenwerk om de juiste waarde in VPK te krijgen. De berekening gaat als volgt:

POKE 23675, waarde - 256 * INT (waarde/256)
POKE 23676, INT (waarde/256)

N.B. U hoeft deze berekening niet in het programma op te nemen omdat u de juiste waarden gemakkelijk met de hand uit kunt rekenen en ze verder gebruiken.

Bijv	Startadres	POKE 23675	POKE 23676
	32600	88	127
	32432	176	126
	32304	48	126

De waarden voor de programmeerbare karakters moet u op de een of andere manier in de computer voeren en dat zijn er dan 464 voor de 58 karakters. Als u deze in DATA-statements zou zetten dan zou dat zoveel ruimte in beslag nemen dat er te weinig overblijft voor de rest van het programma. Het zou trouwens onzinnig zijn om ze elke keer als het programma ingeladen wordt in te moeten voeren. Om dit probleem op te lossen is het programma in twee gedeelten gesplitst. Het eerste gedeelte zet de programmeerbare karakters op en als dit in orde is worden ze als een blok bytes op de cassette gezet. Voordat u het eigenlijke programma gebruikt, wordt dit blok van de cassette gehaald en kan het spelletje gespeeld worden.

N.B. U hoeft het blok alleen maar de eerste keer dat u het programma gebruikt van de cassette te halen. Daarna blijven ze opgeslagen in de computer totdat de stroom wordt afgezet of er andere informatie overheen geschreven wordt.

Het programmeren van de 'snelwegkikker'grafische karakters
Het volgende programma zet de grafische karakters op. Het is belangrijk dat de DATA-statements zorgvuldig ingetikt worden omdat eventuele fouten later moeilijk terug te vinden zijn. Als het programma klaar is dan staan alle data in het VPK-gebied en verschijnen alle gebruikte vormen op het scherm.
Op het scherm moet het volgende te zien zijn:

Bovenaan vier identieke groene kikkers die verschillende kanten op wijzen (omhoog, naar rechts, omlaag en naar links)
Daaronder moet staan:
Een witte motorfiets, een rode auto en een vrachtwagen die allen naar links wijzen
Daaronder een witte motorfiets, een gele auto en een vrachtwagen die naar rechts wijzen.

Als alles in orde is kunt u de karakterset op cassette zetten met de instructie:

SAVE "snelweg-VPK" CODE 32304,464
(U kunt hiervoor elke willekeurige naam gebruiken)

Vergeet niet ook het programma op cassette te zetten voor het geval dat u dat nog een keer nodig heeft.

Het hoofdprogramma 'snelwegkikker'
Het hoofdprogramma is behoorlijk lang, dus is het aan te raden dit in gedeeltes in te tikken en de tekst nauwkeurig te controleren.

Vooral de gedeelten van het programma waarin het scherm opgezet wordt bevatten veel grafische karakters en kleurcontrole-codes die er heel anders uitzien als het programma een keer 'gelopen' heeft (de grafische A bijvoorbeeld is geen 'A' meer als de karakterset ingeladen is). Het is dan bijna onmogelijk geworden om eventuele fouten nog terug te vinden.
Als het programma gestart wordt zal het eerst wachten op het inladen van de karakterset. U moet dan de cassette afdraaien met de 'snelwegVPK'-bytes die u al had opgenomen. Als die bytes er al in zitten en u wilt het programma opstarten dan moet u de instructie RUN 26 gebruiken.

Beschrijving van het programma

Regel	Beschrijving
20 - 25	zet RAMTOP om ruimte te maken voor de grafische karakters en laad ze dan van cassette
26 - 75	initialiseer alle gebruikte variabelen, zet de schermkleur op, de posities van de auto's, vrachtwagens en motoren en de variabelen die gebruikt worden om de VPK karaktersets te kunnen wisselen. (Als de karakterset al aanwezig is, moet het programma hier worden opgestart.)
80 - 100	teken de snelweg
105	teken de kikkers links op het scherm
106	zet SCORE en HOOGSTE SCORE op het scherm
110 - 150	beslis of er een politiewagen over het scherm moet racen. Als dit het geval is dan moet de richting worden bepaald en de wagen op het scherm worden gezet.
500 - 530	controleer of een kikker aangereden is. Zo ja, ga naar regel 4000.
600 - 700	als de speler een zet mag doen (dit is een op de twee keer in de hooldlus) dan wordt het toetsenbord afgetast om te kijken of er een zet gedaan wordt. Zo ja, dan wordt de positie van de kikker veranderd, zo nodig wordt de richting gewijzigd en de score verhoogd als er omhoog of omlaag bewogen is. De oude kikker wordt uitgeveegd en de nieuwe getekend.
750	laat alle motorfietsen bewegen en teken ze
800 - 820	laat een op de twee keer alle auto s bewegen en teken ze
900 - 950	laat één op de vier keer alle vrachtwagens bewegen en teken ze. Spring terug naar regel 106

990 - 999 verwijder een kikker van het scherm.
De positie van de kikker wordt gecontroleerd om te bekijken of snelweg overgetekend moet worden waar de kikker zich bevond.

1000 1070 bestaat uit 4 subroutines om de kikkers in 4 richtingen te kunnen tekenen.

1080 - 1117 6 subroutines om de auto's, motorfietsen en vrachtwagens te tekenen.

2000 - 2001 tekent alle auto's op het scherm.

2010 - 2012 houd de posities van de auto's bij (bij auto's die naar rechts bewegen wordt de positie met 1 vermeerderd, naar links met 1 verminderd).

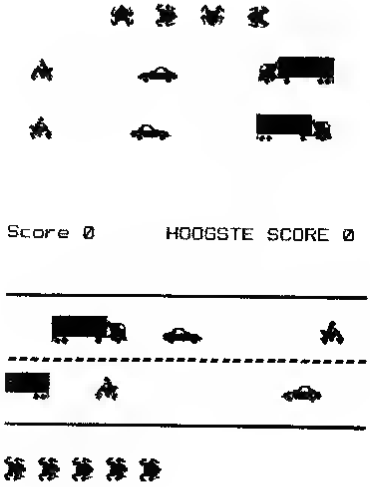
2100 - 2112 teken alle motorfietsen en houd de posities bij.

2200 - 2201 teken alle vrachtwagens.

2210 - 2212 houd de positie van alle vrachtwagens bij.

4000 - 4010 meldt de speler dat er een kikker aangereden is verminder het aantal resterende kikkers met 1. Als er nog kikkers over zijn dan springt het programma terug om met de volgende kikker te beginnen.

4020 - 4070 zet een 'game over'-boodschap op het scherm, pas de HOOGSTE SCORE aan indien nodig en vraag de speler of hij nog een keer wil spelen.



5000 - 5020 als een kikker de thuisbasis haalt springt het programma hier naar toe. Het aantal resterende kikkers wordt met 1 verminderd en als er nog kikkers over zijn dan wordt het spel vervolgd, anders gaat het programma naar regel 4020.

Programma

```
  10 REM snelwegUDG
  20 CLEAR 32303
  25 LET z=0: LET x=256
  30 FOR i=32304 TO 32767
  40 READ a: POKE i,a: NEXT i
  50 BORDER 0: PAPER 0: CLS
  60 POKE 23675,48: POKE 23676,126
  70 PRINT AT 0,9;"EXT SHI 4 GRA AB##EF##IJ##MN EXT SHI 0'
  71 PRINT AT 1,9;"EXT SHI 4 GRA CD##GH##KL##OP EXT SHI 0"
  80 POKE 23675,176
  81 PRINT : PRINT : PRINT "########################## EXT SH
I 5 GRA DDDDC EXT SHI 0'
  82 PRINT "## EXT SHI 7 GRA RS EXT SHI 0 ####### EXT SHI 2
 ## GRA NO EXT SHI 0 ####### EXT SHI 3 GRA LI EXT SHI 5 GRA
 SHI 8 SHI 8 SHI 8 SHI 8 F EXT SHI 0"
  83 PRINT "## EXT SHI 7 GRA TU EXT SHI 0 ####### EXT SHI 2
 GRA ABFQ EXT SHI 0 ####### EXT SHI 3 GRA MK EXT SHI 5 GRA
JGGHF EXT SHI 0"
  90 POKE 23675,88: POKE 23676,127
  91 PRINT : PRINT : PRINT "########################## EXT SHI
5 GRA CDDDD EXT SHI 0"
  92 PRINT "## EXT SHI 7 GRA RS EXT SHI 0 ####### EXT SHI 6
 GRA NO EXT SHI 0 ######### EXT SHI 5 GRA F SHI 8 SHI 8 SHI
 8 SHI 8 EXT SHI 3 GRA JK EXT SHI 0"
  93 PRINT "## EXT SHI 7 GRA TU EXT SHI 0 ####### EXT SHI 6
 GRA POAB EXT SHI 0 ####### EXT SHI 5 GRA EGHHT EXT SHI 3 G
RA LM EXT SHI 0"
 100 PRINT  INK 7;AT 15,0;"Karakters#zijn#geprogrammeerd###
Laad#nu#het#programma#'snelweg'"
 120 STOP
9000 DATA z,1,35,37,111,79,223,x
9001 DATA z,128,196,164,246,242,251,z
9002 DATA 111,15,31,159,220,216,120,48
9003 DATA 246,240,248,249,59,27,30,12
9004 DATA 56,113,193,252,127,31,31,31
9005 DATA 192,240,156,192,240,248,244,254
9006 DATA 31,31,31,127,252,193,113,56
9007 DATA 254,244,248,240,192,156,240,192
9008 DATA 48,120,216,220,159,31,15,111
9009 DATA 12,30,27,59,249,248,240,246
9010 DATA x,223,79,111,47,35,1,z
9011 DATA x,251,242,246,164,196,128,z
9012 DATA 3,15,57,3,15,31,47,127
9013 DATA 28,142,131,63,254,248,240,240
9014 DATA 127,47,31,15,3,57,15,3
9015 DATA 240,240,248,254,63,131,142,28
```

```
9016 DATA z,z,3,7,15,2,z,z
9017 DATA 7,x,x,159,111,247,240,96
9018 DATA z,z,z,z,z,254,254,254
9019 DATA z,z,z,z,z,x,x,x
9020 DATA 254,254,254,254,254,254,254,254
9021 DATA 254,254,254,4,50,122,122,48
9022 DATA x,x,x,z,z,z,z,z
9023 DATA x,x,x,z,6,15,15,6
9024 DATA 2,2,250,250,254,252,252,248
9025 DATA x,x,x,x,6,15,15,6
9026 DATA 248,252,254,127,184,216,192,128
9027 DATA z,z,7,9,17,17,31,31
9028 DATA 31,31,31,62,61,59,3,1
9029 DATA z,z,z,z,z,63,97,193
9030 DATA z,z,z,z,z,z,128,192
9031 DATA 128,x,x,x,x,254,z,z
9032 DATA 240,254,x,159,111,246,240,96
9033 DATA 1,3,1,z,3,4,14,31
9034 DATA 128,192,192,224,224,112,119,255
9035 DATA 31,63,115,81,169,112,112,32
9036 DATA 254,252,252,234,213,206,14,4
9037 DATA 224,x,x,249,246,239,15,6
9038 DATA z,z,192,224,240,64,z,z
9039 DATA z,z,z,z,z,127,127,127
9040 DATA z,z,z,z,z,z,x,x
9041 DATA 127,127,127,32,76,94,94,12
9042 DATA 127,127,127,127,127,127,127,127
9043 DATA x,x,x,z,96,240,240,96
9044 DATA x,x,x,z,z,z,z,z
9045 DATA x,x,x,x,96,240,240,96
9046 DATA 64,64,95,95,127,63,63,31
9047 DATA z,z,224,144,136,136,248,248
9048 DATA 31,63,127,254,79,27,3,1
9049 DATA 248,248,248,124,188,220,192,128
9050 DATA z,z,z,z,z,z,1,3
9051 DATA z,z,z,z,z,252,134,131
9052 DATA 15,127,x,249,246,111,15,6
9053 DATA 1,x,x,x,x,127,z,z
9054 DATA 1,3,3,7,7,14,238,x
9055 DATA 128,192,128,z,192,32,112,248
9056 DATA 127,63,63,87,171,115,113,32
9057 DATA 248,252,206,138,149,14,14,4
```

Programma

```
10 REM snelweg
20 CLEAR 32500: RANDOMIZE
25 REM  LOAD ""CODE
26 LET h1=0
30 LET m1=2*INT RND*15: LET m2=2*INT RND*15
33 LET t1=INT RND*30: LET t2=INT RND*30
34 LET c1=2*INT RND*30: LET c2=2*INT RND*30
35 LET ce=2: LET cf=6
36 LET pz=4: LET tz=4: LET mz=2
```

```
  40 BORDER 0: PAPER 0: INK 7: CLS
  45 LET f=4: LET fc=5
  50 LET ul=23675: LET uh=23676
  55 LET sc=0
  60 LET l1=88: LET l2=176: LET l3=48
  70 LET h1=127: LET h2=126
  75 LET l10=8: LET l11=9: LET l12=7: LET l20=13: LET l21=1
4: LET l22=12
  80 PLOT 0,38: DRAW 255,0: DRAW 0,1: DRAW -255,0
  90 PLOT 0,128: DRAW 255,0: DRAW 0,1: DRAW -255,0
 100 FOR x=2 TO 256 STEP 8: PLOT x,83: DRAW 4,0: DRAW 0,1:
DRAW -4,0: NEXT x
 105 LET pa=20: LET py=20: FOR x=0 TO (fc-1)*3 STEP 3: LET
px=x: GO SUB 1020: NEXT x
 106 PRINT AT 0,0;" EXT SHI 6 Score#EXT SHI 7";sc;AT 0,9:"E
XT SHI 6 HOOGSTE#SCORE# EXT SHI 7";hi;'EXT SHI 0"
 110 IF RND<.9 THEN  GO TO 500
 115 IF RND<.5 THEN  GO TO 150
 120 LET c=5: FOR j=1 TO 6: BEEP .2,10: BEEP .2,20: NEXT j:
FOR x=-4 TO 30 STEP 2: GO SUB 1080: NEXT x
 130 GO TO 500
 150 LET c=5: FOR j=1 TO 6: BEEP .2,20: BEEP .2,10: NEXT j:
FOR x=30 TO -4 STEP -2: GO SUB 1140: NEXT x
 500 IF py>15 OR (py<12 AND py>9) OR py<6 THEN  GO TO 600
 510 LET a=22528+py*32+px
 520 IF PEEK a<>4 OR PEEK (a+1)<>4 THEN  GO TO 4000
 530 IF PEEK (a+32)<>4 OR PEEK (a+33)<>4 THEN  GO TO 4000
 600 IF pz THEN  LET pz=pz-1: GO TO 700
 610 LET pz=1: LET z$=INKEY$: IF z$="" THEN  GO TO 700
 615 GO SUB 990
 620 LET py=py-(z$="q")+(z$="a")
 630 LET px=px+(z$="p")-(z$="o")
 640 IF z$="q" THEN  LET pa=0
 641 IF z$="p" THEN  LET pa=20
 642 IF z$="a" THEN  LET pa=40
 643 IF z$="o" THEN  LET pa=60
 650 IF py=0 THEN  GO SUB 5000
 660 LET sc=sc+10*(z$="q" OR z$="a")
 700 GO SUB 1000+pa
 750 GO SUB 2100
 800 IF mz THEN  LET mz=mz-1
 810 GO SUB 2000
 820 IF NOT mz THEN  LET mz=2: GO SUB 2010
 900 IF tz THEN  LET tz=tz-1
 901 GO SUB 2200
 902 IF NOT tz THEN  GO SUB 2210: LET tz=4
 950 GO TO 106
 990 PRINT AT py,px;"##";AT py+1,px;"##":
 991 IF py=10 OR py=11 THEN  PLOT px*8+2,83: DRAW 4,0: DRAW
 0,1: DRAW -4,0: PLOT px*8+10,83: DRAW 4,0: DRAW 0,1: DRAW
-4,0: RETURN
 995 IF py=17 OR py=16 THEN  PLOT px*8,78: GO TO 998
 996 IF py<>5 AND py<>4 THEN  RETURN
 997 PLOT px*8,128
 998 DRAW 16,0: DRAW 0,1: DRAW -16,0
```

```
 999 RETURN
1000 POKE u1,13: POKE uh,h2
1010 PRINT  INK f;AT py,px;"GRA AB GRU";AT py+1,px;"GRA CD
GRU": RETURN
1020 POKE u1,13: POKE uh,h2
1030 PRINT  INK f;AT py,px;"GRA EF GRU";AT py+1,px;"GRA GH
GRU": RETURN
1040 POKE u1,13: POKE uh,h2
1050 PRINT  INK f;AT py,px;'GRA IJ GRU";AT py+1,px;"GRA KL
GRU": RETURN
1060 POKE u1,13: POKE uh,h2
1070 PRINT  INK f;AT py,px;'GRA MN GRU";AT py+1,px;"GRA UP
GRU": RETURN
1080 POKE u1,11: POKE uh,h1
1085 IF x<0 THEN  PRINT AT 110,0; INK c;"## GRA NO GRU ##"(
1-x TO );AT 111,0;"GRA PQPQAB GRU'(1-x TO ): RETURN
1090 IF x>26 THEN  PRINT AT 110,x;"EXT SHI 0 ##"; INK c;'GR
A NO GRU"( TO 30-x);AT 111,x;"EXT SHI 0 ##"; INK c;"GRA PQ
GRU"( TO 30-x): RETURN
1095 PRINT AT 110,x;"EXT SHI 0 ##"; INK c;"GRA NO GRU ##";A
T 111,x;'EXT SHI 0 ##"; INK c;"GRA PQAB GRU": RETURN
1100 POKE u1,11: POKE uh,h1
1104 IF x>=0 AND x<25 THEN  PRINT AT 112,x;"EXT SHI 0 # EXT
 SHI 5 GRA CDDDD GRU";AT 110,x;"EXT SHI 0 # EXT SHI 5 GRA F
 SHI 8 SHI 8 SHI 8 SHI 8 EXT SHI 3 GRA JK GRU";AT 111,x;"EX
I SHI 0 # EXT SHI 5 GRA EGHHI EXT SHI 3 GRA LM GRU": RETURN

1108 IF x<-5 THEN  PRINT "EXT SHI 3";AT 110,0;"GRA JK GRU"(
-5-x TO );AT 111,0;'GRA LM GRU"(-5-x TO ): RETURN
1112 IF x<0 THEN  PRINT AT 112,0;"EXT SHI 5";"GRA CDDDD GRU
"(-x TO );AT 110,0;' GRA F SHI 8 SHI 8 SHI 8 SHI 8 GRU"(-x
TO );"EXT SHI 3 GRA JK GRU";AT 111,0;'EXT SHI 5";"ORA EGHHI
 GRU"(-x TO );'EXT SHI 3 GRA LM GRU": RETURN
1115 IF x=25 THEN  PRINT AT 112,x;'EXT SHI 0 # EXT SHI 5 GR
A CDDDD GRU";;AT 110,x;"EXT SHI 0 # EXT SHI 5 GRA F SHI 8 S
HI 8 SHI 8 SHI 8 EXT SHI 3 GRA J GRU";AT 111,x;"EXT SHI 0 #
 EXT SHI 5 GRA EGHHI EXT SHI 3 GRA L GRU": RETURN
1117 PRINT AT 112,x;"EXT SHI 0 # EXT SHI 5";"GRA CDDDD GRU"
( TO 31-x);AT 110,x;"EXT SHI 0 # EXT SHI 5";"GRA F SHI 8 SH
I 8 SHI 8 SHI 8 GRU"( TO 31-x);AT 111,x;"EXT SHI 0 # EXT SH
I 5";"GRA EGHHI GRU'( TO 31-x): RETURN
1120 POKE u1,11: POKE uh,h1
1125 IF x<0 THEN  PRINT AT 110,0; INK 7;"GRA RS GRU";AT 111
,0; INK 7;'GRA TU GRU": RETURN
1130 IF x>28 THEN  PRINT AT 110,x;"EXT SHI 0 ##";AT 111,x;"
EXT SHI 0 ##": RETURN
1135 PRINT AT 110,x;"EXT SHI 0 ## EXT SHI 7 GRA RS EXT SHI
0";AT 111,x;'EXT SHI 0 ## EXT SHI 7 GRA TU EXT SHI 0": RETU
RN
1140 POKE u1,12: POKE uh,h2
1145 IF x<26 THEN  PRINT AT 120,x; INK c;"## GRA NO GRU"( T
O 32-x);AT 121,x; INK c;"GRA ABPQ GRU"( TO 32-x): RETURN
1150 IF x<0 THEN  PRINT AT 120,0; INK c;'# GRA NO EXT SHI 0
 ##"(-x TO );AT 121,0; INK c;'GRA BPQ EXT SHI 0 ##"(-x TO )
: RETURN
```

```
1155 PRINT AT 120,x; INK c;"## GRA NO EXT SHI 0 ##";AT 121,
x; INK c;"GRA ABP@ EXT SHI 0 ##": RETURN
1160 POKE u1,l2: POKE uh,h2
1163 IF x>=0 AND x<25 THEN  PRINT AT 122,x;'EXT SHI 0 ## EX
T SHI 5 GRA DDDDC EXT SHI 0 #';AT 120,x;"EXT SHI 3 GRA LIEX
T SHI 5 GRA SHI 8 SHI 8 SHI 8 SHI 8 E EXT SHI 0 #";AT 121,x
;"EXT SHI 3 GRA MK EXT SHI 5  GRA JGGHF EXT SHI 0 #": RETUR
N
1165 IF x=-1 THEN  PRINT AT 122,0;"EXT SHI 0 # EXT SHI 5 GR
A DDDDC EXT SHI 0 #";AT 120,0;'EXT SHI 3 GRA I EXT SHI 5 GR
A SHI 8 SHI 8 SHI 8 SHI 8 E EXT SHI 0 #";AT 121,0;"EXT SHI
3 GRA K EXT SHI 5 GRA JGGHF EXT SHI 0 #": RETURN
1170 IF x<-1 THEN  PRINT AT 122,0;"EXT SHI 5";'# GRA DDDDC
EXT SHI 0 #"(-x TO );AT 120,0;"EXT SHI 5";"# GRA SHI 8 SHI
8 SHI 8 SHI 8 E EXT SHI 0 #'(-x TO );AT 121,0;'EXT SHI 5";"
# GRA JGGHF EXT SHI 0 #"(-x TO ): RETURN
1173 IF x>29 THEN  PRINT AT 120,x;"EXT SHI 3";'GRA LI GRU"(
 TO 32-x);AT 121,x;"GRA MK GRU"( TO 32-x): RETURN
1175 PRINT AT 122,x;"EXT SHI 0 ## EXT SHI 5";"GRA DDDDC GRU
"( TO 30-x);AT 120,x;"EXT SHI 3 GRA LI EXT SHI 5';"GRA SHI
8 SHI 8 SHI 8 SHI 8 E GRU"( TO 30-x);AT 121,x;'EXT SHI 3 GR
A MK EXT SHI 5";'GRA JGGHF GRU"( TO 30-x): RETURN
1180 POKE u1,l2: POKE uh,h2
1185 IF x>29 THEN  PRINT AT 120,x; INK 7;"GRA RS GRU";AT 12
1,x;"GRA TU GRU": RETURN
1190 IF x<0 THEN  PRINT AT 120,0; INK 7;'GRA S EXT SHI 0 ##
"(-x TO );AT 121,0; INK 7;"GRA U EXT SHI 0 ##"(-x TO ): RET
URN
1195 PRINT AT 120,x;"EXT SHI 7 GRA RS EXT SHI 0 ##";AT 121,
x;"EXT SHI 7 GRA TU EXT SHI 0 ##": RETURN
2000 LET c=ce: LET x=c1: GO SUB 1080
2001 LET c=cf: LET x=c2: GO TO 1140
2010 LET c1=c1+2: IF c1=32 THEN  LET c1=-4
2011 LET c2=c2-2: IF c2=-6 THEN  LET c2=30
2012 RETURN
2100 LET x=m1: GO SUB 1120
2101 LET m1=m1+2: IF m1=32 THEN  LET m1=-2
2110 LET x=m2: GO SUB 1180
2111 LET m2=m2-2: IF m2=-4 THEN  LET m2=30
2112 RETURN
2200 LET x=t1: GO SUB 1100
2201 LET x=t2: GO TO 1160
2210 LET t1=t1+1: IF t1=32 THEN  LET t1=-7
2211 LET t2=t2-1: IF t2=-8 THEN  LET t2=31
2212 RETURN
4000 LET f=2: FOR i=50 TO 1 STEP -5: GO SUB 990: GO SUB 100
0+pa: BEEP .02,i: NEXT i
4005 GO SUB 990: LET f=4
4010 LET fc=fc-1: IF fc<0 THEN  GO TO 105
4020 PRINT AT 1,10;"Game#over"
4030 IF sc>hi THEN  LET hi=sc: PRINT AT 2,3;"Maar#je#score#
is#de#nieuwe########HOOGSTE##SCORE'
4040 PRINT AT 4,6;"Toets#'j'#voor#nieuw#spel'
4050 IF INKEY$="" THEN  GO TO 4050
4060 IF INKEY$="j" THEN  GO TO 30
```

```
4070 STOP
5000 FOR i=1 TO 50 STEP .5: GO SUB 990: GO SUB 1000+pa: BEEP
 .02,i: NEXT i
5005 GO SUB 990
5010 LET fc=fc-1: IF fc<0 THEN  GO TO 105
5020 GO TO 4020
```

6. Hulpprogramma's

6.1 Vrij programmeerbare karakters
(Copyright Beam Software)

Beschrijving
Dit programma is leuk om mee te spelen. U krijgt de vrij programmeerbare karakters sterk vergroot op het scherm te zien. Maar het programma is ook een absolute 'must' voor programmeurs die de grafische karakters om willen zetten in High resolution-karakters voor eigen gebruik, bijvoorbeeld voor spelletjes. Een zeer krachtige 'editor' is ingebouwd met de volgende mogelijkheden.

ontwerpen van karakters wordt visueel ondersteund
karakters kunnen onmiddellijk worden leeggemaakt
karakters kunnen onmiddellijk worden gevuld
vergrote weergave van karakters op scherm
karakters kunnen worden gespiegeld
cursor loopt in vier richtingen door
optie om 1, 2 of 4 karakters tegelijk te programmeren
gemakkelijk te identificeren kleuren
elk moment ophouden zonder veranderingen

En dan wordt nog de inhoud van de geheugenadressen van deze karakters automatisch aangepast na het gebruik van de 'editor' en alle gegevens die op deze karakters van toepassing zijn worden op het scherm gezet met de mogelijkheid om ze naar de printer te sturen.
U zult verrast zijn hoe handig dit programma kan zijn. Terwijl het programma zijn werk doet wordt u voortdurend op de hoogte gehouden van waar hij mee bezig is.

Gebruik van het 'editor'-programma:

cursor-besturing
5 naar links
6 omlaag
7 omhoog
8 naar rechts
1 punt onder cursor vullen
0 punt onder cursor leegmaken

routine-besturing
b alle punten leegmaken
t alle punten vullen
m de karakterblokken intern spiegelen
s sla de aangepaste blokken op in geheugen
E Einde veranderingen, verlaat 'editor' zonder blokken te veranderen (N.B. shift 'e !)

Programma-opbouw

Het programma gebruikt een array om de geheugenbytes na te bootsen die een karakterblok definiëren. Op een 48K-Spectrum begint het karakterblok van het eerste vrij programmeerbare karakter op adres 65368, op een 16K-Spectrum is dit adres 32600. Daarna volgen 8 bytes die de vorm van het karakter definieren.

De opbouw van het programma is als volgt:

```
Invoer
  definieer schermvariabelen
  invoer begin karakterblok
  invoer aantal te programmeren karakters (1, 2 of 4)
  definieer dimensie van de volgende arrays: simulatie, geheugenadres-wijzer, spiegel

Hoofdlus
  initialiseer simulatie, geheugenadres-wijzer array
  schrijf geheugenbytes in simulatie-array
  zet vergrote en normale vorm karakters op scherm
  voer 'Editor'-routine uit
  tot 's'-sla op of 'E'-einde
  zet printeroptie op scherm
  als andere karakters veranderd moeten worden, ga dan naar Invoer

Editor-routine
  bepaal afmetingen vergroot blok
E1: flitscursor
  accepteer invoer besturingstoetsen
  flitscursor uit
  als Invoer besturingstoets is:

    '5' of '8'    bereken nieuwe cursorpositie
                  ga naar E1
    '0'           zet huidig simulatie-array-element op 0, pas scherm aan
                  ga naar E1
    '1'           zet huidig simulatie-array-element op 1, pas scherm aan
                  ga naar E1
    'f' of 'b'    als 'f' zet hele simulatie-array op 1,
                  als 'b' zet hele simulatie-array op 0
                  zet karakterblok opnieuw op scherm
                  ga naar E1
    's'           zet printeroptie op scherm
                  als ja dan sla simulatie-array op in
                  geheugen
                  als nee ga dan naar E1
    'm'           spiegel simulatie-array-regel
                  ga naar E1
    'E'           return
```

Opmerkingen
Het aan- en uitknipperen van de cursor wordt bereikt door de attributen van de karakterpositie in de 'display file' (het geheugengebied van het scherm) te veranderen. Het geheugenadres van de attributen van het karakter links boven in het scherm is 22528. Als u nu een karakter op regel x, kolom y wilt laten knipperen, dan bereikt u dat door:
3040 LET a = 22528 + 32 * x + y: POKE a, PEEK a + 128
Op dezelfde wijze kunt u het knipperen laten stoppen door:
3060 POKE a, PEEK a - 128

Om met een COPY-opdracht het scherm op de printer te kunnen zetten krijgt een lege punt op het scherm de attributen PAPER yellow, INK BLACK, en een gevulde punt PAPER white, INK BLUE.

Om het karakter vergroot op het scherm te krijgen gebruikt u een PRINT-opdracht, voor een afdruk op normale grootte een PLOT-opdracht.

Om het juiste vrij programmeerbare karakter aan te kunnen roepen vertaalt u een INPUT van kleine letter naar een hoofdletter door:
8060 LET K$ = CHR$ (CODE K$ - 32)

Er wordt ook een controle uitgevoerd op de INPUT van een letter om te kijken of die soms buiten het bereik van de vrij programmeerbare karakter-set ligt:
8070 ... OR (CODE K$ + nc + 79 -1) > 164
m.a.w. de ASCII-code van de eerste hoofdletter + het aantal te programmeren karakters + 79 -1 mag niet groter zijn dan 164, de code van het laatste vrij programmeerbare karakter.

Om het beginadres te vinden van de 8 geheugenbytes die een bepaald vrij programmeerbaar karakter definieren gebruiken we: 8150 LET S(I + 1) = USR CHR$ (CODE K$ + I)

De logica achter het intern spiegelen van een array berust op het opdelen van de array in twee arrays waarna de elementen van buiten naar binnen toe worden verwisseld. Een reeks van 8 elementen zal dus in twee helften van ieder 4 elementen worden verdeeld. Het eerste element van de eerste helft zal daarna omgeruild worden met het laatste element van de tweede helft. Daarna wordt het tweede element geruild met het een na laatste. Dit gaat door totdat het laatste element van de eerste helft is verwisseld met het eerste element van de tweede helft. Deze routine staat in regel 3330 to 3400

Programma

```
 100 REM
 110 REM vrij programmeerbare     karakters
 115 REM
 120 GO TO 8000
1000 REM teken karakterset
1010 REM Input r,c;start coordinaten
1020 PRINT AT 8,11;"##": PRINT AT 9,11;"##": FOR H=1 TO nc
1030 LET hr=8*(H=3 OR H=4): LET hc=8*(H=2 OR H=4)
1040 FOR I=1 TO 8
1050 FOR J=1 TO 8
1060 PRINT AT (I+r+hr),(J+c+hc);
1070 IF B(I+hr,J+hc)=0 THEN  PRINT  PAPER 6; INK 0;"#": GO
TO 1090
1080 PRINT  PAPER 7; INK 1;"GRA SHI 8 CRU": PLOT 87+J+hc,11
2-I-hr
1090 NEXT J
1100 NEXT I
1110 NEXT H
1120 LET L$="##################################"
1130 RETURN
2000 REM
2010 REM karakter in array
2020 REM Input K$,Output B(I,J)
2030 REM
2040 FOR H=1 TO nc
2050 LET hr=8*(H>2): LET hc=8*(H=2 OR H=4)
2060 FOR I=1 TO 8
2070 LET m=PEEK  (s(H)+I-1)
2080 FOR J=8 TO 1 STEP -1
2090 LET B(I+hr,J+hc)=m-2*INT  (m/2)
2100 LET m=INT  (m/2)
```

```
2110 NEXT J
2120 NEXT I
2130 NEXT H
2140 RETURN
3000 REM Edit KARAKTER,x=regel,y=kolom
3010 REM Input B(I,J)
3020 LET hr=8*(nc>2): LET hc=8*(nc=2 OR nc=4)
3030 LET x=r+1: LET y=c+1
3040 LET a=22528+32*x+y: POKE a,PEEK a+128: PAUSE 200
3050 IF INKEY$="" THEN GO TO 3050
3060 LET M$=INKEY$: POKE a,PEEK a-128: REM unflash oude pos
itie
3070 IF M$<"5" OR M$>"8" THEN GO TO 3150
3080 LET y=y+(M$="8" AND (x<8+hr OR y<22+hc))-(M$="5" AND
 (x<>1 OR y<>c+1))
3090 LET x=x+(M$="6")-(M$="7")
3100 IF x>8+hr THEN LET x=1
3110 IF x<1 THEN LET x=8+hr
3120 IF y>22+hc THEN LET y=c+1: LET x=x+1
3130 IF y<15 THEN LET y=22+hc: LET x=x-1
3140 GO TO 3040
3150 IF M$<>"0" AND M$<>"1" THEN GO TO 3190
3160 PRINT AT x,y;
3170 IF B(x-r,y-c)=1 AND M$="0" THEN LET B(x-r,y-c)=0: PRI
NT PAPER 6; INK 0;"#": PLOT OVER 1;87+y-c,112-x+r: GO TO
3040
3180 IF B(x-r,y-c)=0 AND M$="1" THEN LET B(x-r,y-c)=1: PRI
NT PAPER 7; INK 1;"GRO SHI FT GNU": PLOT 87+y-c,112-x+r: GO
 TO 3040
3190 IF M$<>"f" AND M$<>"F" AND M$<>"b" AND M$<>"B" THEN G
O TO 3240
3200 LET b=(M$="f" OR M$="F")
3210 GO SUB 4000: REM init ARRAY
3220 GO SUB 1000: REM teken KAR
3230 GO TO 3030: REM nieuw begin
3240 IF M$<>"E" THEN GO TO 3260
3250 PRINT AT 19,0; INK 2;"DIT#KARAKTER#BLIJFT#ONVERANDERD'
": RETURN
3260 IF M$<>"s" AND M$<>"S" THEN GO TO 3320: REM edit inke
y
3270 PRINT AT 18,0;"Dit#opslaan#(j#of#n)#";
3280 LET e$=INKEY$: IF e$<>"j" AND e$<>"n" THEN GO TO 3280

3290 PRINT INK 2;e$: PAUSE 50
3300 IF e$="j" THEN GO SUB 5000: RETURN
3310 FOR M=1 TO 3: PRINT AT 17+M,0;: PRINT "################
##################";: NEXT M: GO TO 3030
3320 IF M$<>"m" AND M$<>"M" THEN GO TO 3040
3330 PRINT AT 0,25; FLASH 1; INVERSE 1; INK 3;"SPIEGEL": LE
T fc=INT (bc/2+.5)
3340 FOR I=1 TO br
3350 FOR J=1 TO fc
3360 LET M=B(I,J)
3370 LET B(I,J)=B(I,bc-J+1)
3380 LET B(I,bc-J+1)=M
```

```
3390 NEXT J
3400 NEXT I
3410 PRINT AT 0,25;"#######"
3420 GO SUB 1000
3430 GO TO 3040
4000 REM Initialiseer simulatie ARRAY
4010 FOR I=1 TO br
4020 FOR J=1 TO hc
4030 LET B(I,J)=b
4040 NEXT J
4050 NEXT I
4060 RETURN
4070 REM
5000 REM verplaats array naar   geheugen
5010 PRINT AT 0,25; FLASH 1; PAPER 3; INK 7;"SAVING"
5020 FOR H=1 TO nc
5030 LET hr=8*(H>2): LET hc=8*(H=2 OR H=4)
5040 FOR I=1 TO 8
5050 LET t=0
5060 FOR J=8 TO 1 STEP -1
5070 IF B(I+hr,J+hc)=1 THEN  LET t=t+2^(8-J)
5080 NEXT J
5090 POKE (s(H)+I-1),t
5100 NEXT I
5110 NEXT H
5120 PRINT AT 0,25;"######"
5130 PRINT AT 0,25; PAPER 3; INK 7;"SAVED!!"
5140 RETURN
6000 REM
6010 REM Lees geheugenwaarde van karakter
6020 PRINT AT 0,0;"########"
6030 FOR H=1 TO nc
6040 LET hc=(H=2 OR H=4)*5+1: LET hr=(H>2)*9
6050 PRINT AT hr,hc-1; PAPER 1; INK 7;"##";CHR$ (CODE k$+H-
1);"#"
6060 FOR I=1 TO 8
6070 PRINT AT hr+I,hc;PEEK (s(H)+I-1)
6080 NEXT I
6090 NEXT H
6100 LET UG=CODE k$+79: PRINT AT 8,11; OVER 0;CHR$ UG;
6110 IF nc>=2 THEN  PRINT CHR$ (UG+1)
6120 IF nc=4 THEN  PRINT AT 9,11; OVER 0;CHR$ (UG+2);CHR$ (
UG+3)
6130 RETURN
8000 BORDER 5: INK 0: PAPER 7: OVER 0: FLASH 0: CLS
8010 REM toets repre van KAR: DIM l$(1)
8020 PRINT "Welk#karakter#programmeren#?": PRINT "(A#-#U)"
8030 LET k$=INKEY$: IF k$>="A" AND k$ <="U" THEN  GO TO 8060

8040 IF k$<"a" OR k$>"u" THEN  GO TO 8030
8050 LET k$=CHR$ (CODE k$-32)
8060 PRINT K$: PRINT : PRINT "Hoeveel#karakters#";"##(1#-#4)#
";
8070 LET nc=((CODE INKEY$)-48): IF nc<1 OR nc>4 OR (CODE k$
+nc+79-1)>164 THEN  GO TO 8070
```

```
8080 IF nc=3 THEN  LET nc=4
8090 PRINT nc: FOR I=1 TO 50: NEXT I: CLS
8100 DIM S(nc): REM geheugen adres karakterset
8110 LET br=8*(1+(nc=4)): LET bc=8*(1+(nc=1))
8120 DIM C(bc): DIM B(br,bc): REM simulatie array
8130 FOR I=0 TO nc-1
8140 PRINT  INK 2;CHR$ (CODE K$+I);"#";: REM print karakter
s
8150 LET S(1+I)=USR CHR$ (CODE K$+I): REM geh adres kar set

8160 NEXT I
8170 PRINT AT 0,25; FLASH 1; PAPER 1; INK 7;"LOADING"
8180 LET b=0: GO SUB 4000: REM initialiseer ARRAY
8190 GO SUB 2000: REM kar naar array
8200 LET r=0: LET c=14
8210 GO SUB 1000: REM Haal KAR terug
8220 PRINT AT 0,25;"########"
8230 GO SUB 3000: REM Edit KAR
8240 GO SUB 6000: REM list geheugenwaarde
8250 PRINT AT 20,0;"Copie#op#printer?#(j#of#n)";
8260 LET g$=INKEY$: IF g$ <> "j" AND g$ <> "n" THEN  GO TO 8260

8270 PRINT g$
8280 IF g$="n" THEN  GO TO 8320
8290 PRINT AT 21,0;"Willekeurige#toets#als#je#klaar#bent":
PAUSE 2000
8300 IF INKEY$="" THEN  GO TO 8300
8310 COPY
8320 PRINT AT 20,0;L$: PRINT AT 21,0;L$
8330 PRINT AT 21,0;L$
8340 PRINT AT 20,0;"Andere#karakters#veranderen#";"###(j#of#n
)";: PAUSE 200
8350 LET g$=INKEY$: IF g$ <> "j" AND g$ <> "n" THEN  GO TO 8350

8360 PRINT g$
8370 IF g$="n" THEN  STOP
8380 CLS : RUN
```

6.2 Regelhernummering

(Copyright Beam Software)

Beschrijving

Hier volgen een paar programma's die nuttig kunnen zijn als het geregeld voorkomt dat u een programma ontwikkeld heeft en dan merkt dat de regelnummering een chaos is en geen ruimte meer toelaat om die geniale subroutine in te voegen die u net had bedacht

Model 1 (hernum1) is een kort hernummerprogramma dat automatisch alle regels tussen regel 100 en 9990 (waar model 1 begint) hernummert in stappen van 10.

Model 2 (hernum2) is een kort programma met dezelfde specificaties als model 1 behalve dat het ook alle GOTO en GOSUB-instructies in een regel op het scherm zet, vergezeld van het nieuwe regelnummer.

De GOTO en GOSUB-instructies worden zelf niet aangepast omdat bij de ZX-Spectrum deze instructies ook gebruikt kunnen worden met een variabele in plaats van een getal. Het zou te moeilijk zijn voor het programma om deze ook te berekenen. De beste oplossing is om op het scherm aan te geven waar ze staan in de hernummerde versie zodat u ze zelf kunt aanpassen.

Model 3 (hernum3) is de lange versie van het hernummerprogramma. Het kan een blok regels hernummeren in elke stapgrootte en zet de GOTO en GOSUB-instructies op het scherm. Het GOTO en GOSUB-rapport kunt u naar de printer sturen, maar het belangrijkste is dat het programma compleet Menu-gestuurd is met grote nadruk op de 'gebruikersvriendelijkheid'.

Om een van deze modellen te kunnen gebruiken moet u ze met een MERGE inladen in plaats van met een LOAD, omdat anders het programma dat u wilt hernummeren uit het geheugen zou worden geveegd.
Een regel in een BASIC-programma wordt als volgt in de SPECTRUM in het geheugen opgeslagen:

2 bytes: regelnummer met meest gehanteerde byte voorop
2 bytes: regellengte met minst gehanteerde byte voorop
n bytes: inhoud van de regel inclusief 'end of line'-byte.

Het BASIC-programma begint op adres 23755 als er geen Microdrives zijn aangesloten.

Model 1 hernummeren
Gebruik RUN 9990 of GOTO 9990 om het programma op te starten.

Programma-opbouw

```
  zet NIEUW regel # op 100
  zet geheugenpointer op 23755
L1: als oud regel # >= 9990 dan STOP
  plaats NIEUW regel # in geheugen
  zoek verhoogd NIEUW regel #
  zet geheugenpointer op volgende regel
  ga naar L1;
```

Opmerkingen
Om het hernummerprogramma uit het geheugen te verwijderen na gebruik zou u het Regelblok verwijder-programma elders in het boek kunnen gebruiken.
Dit hernummerprogramma beslaat regel 9990 - 9995.

Model 2 hernummeren
Gebruik RUN 9988 of GOTO 9988 om het programma op te starten.

Programma-opbouw

```
  initialiseer NIEUW regel #, stapgrootte, begin van blok en eind van blok
  zoek geheugenadres begin van blok
L1: als eind van blok bereikt dan STOP
  POKE NIEUW regel in geheugen
  rapporteer een eventuele GOTO en GOSUB in deze regel
  pas geheugenpointer en regel # aan
  ga naar L1
```

Opmerkingen
Daar dit programma een ingebouwde code heeft om het geheugenadres van het begin van een blok te vinden kunt u de namen en waarden van de variabelen in regel 9988 aanpassen aan uw eigen wensen. Ook als u dit korte programma wilt verwijderen dan moet u dat met de hand doen of het programma Regelblok verwijder gebruiken.
Als u het GOTO en GOSUB-rapport op de printer wilt zetten, dan zult u dat bij dit programma nog met de hand moeten doen.

Dit hernummerprogramma beslaat regel 9988 - 9999.

Model 3 hernummeren
Gebruik RUN 9900 of GOTO 9900 om het programma op te starten.

Programma-opbouw
In dit programma worden technieken toegepast om met het scherm te communiceren, om het eerste en laatste regelnummer van het te hernummeren blok, de stapgrootte en de nieuwe beginregel van het blok bij de gebruiker op te vragen.
Alle opgevraagde waarden worden tot een integer getal afgerond.

De opbouw is als volgt.

```
Invoer
  begin- en eindregel van het blok
  stapgrootte en nieuwe beginregel van het blok
Zoek geheugenadres beginregel
CALL HERNUMMER module (gelijk aan model 2)
OPTIE VERVOLG met HERNUMMER ander blok
  of COPIE GOTO en GOSUB-rapport naar printer
  of STOP stop of verwijder model 3 uit geheugen
```

Opmerkingen
Dit programma biedt u de mogelijkheid zichzelf uit het geheugen te verwijderen na gebruik.

Het hernummeren geschiedt inclusief de blokregels.
Het programma beslaat regel 9900 - 9999.
Als u de optie kiest om het programma zichzelf te laten verwijderen dan moet u na afloop

9900 intikken, gevolgd door ENTER omdat het programma zichzelf in een lang REM-statement zet op regel 9900.
Het programma kan ook te allen tijde worden verwijderd door GOTO 9945 in te tikken.

Een interessante toepassing van dit programma is het vastleggen van een bepaalde boodschap in een REM-statement bijvoorbeeld

```
10 REM Copyright Beam Software
```

U kunt deze regel dan hernummeren tot regel 0 door als nieuwe beginregel 0 op te geven. Na hernummering is deze regel niet meer te veranderen of te verwijderen tenzij het regelnummer weer veranderd wordt in een getal groter dan 0.

Programma

```
9990 REM HERNUMMER model 1
9991 LET i=100: LET n=23755
9992 IF (256*PEEK n+PEEK (n+1))>=9990 THEN  STOP
9993 POKE n,INT (i/256): POKE (n+1),i-INT (i/256)*256
9994 LET i=i+10: LET n=n+3+PEEK (n+2)+256*PEEK (n+3)+1
9995 GO TO 9992
```

Programma

```
9987 REM HERNUMMER model 2
9988 LET NIEUW=100: LET STAP=10: LET BVB=100: LET EVB=9980
9989 LET i=NIEUW: LET n=23755
9990 IF (256*PEEK n+PEEK (n+1))>=BVB THEN  GO TO 9992
9991 LET n=n+3+PEEK (n+2)+256*PEEK (n+3)+1: GO TO 9990
9992 LET N1=(256*PEEK n+PEEK (n+1)): IF N1>EVB THEN  STOP
9993 POKE n,INT (i/256)
9994 POKE (n+1),i-256*INT (i/256)
9995 FOR I=N+4 TO N+3+PEEK (N+2)+256*PEEK (N+3)
9996 IF PEEK I=236 OR PEEK I-237 THEN  PRINT L;"-";CHR$ PEE
K I
9997 NEXT I
9998 LET N=I: LET L=L+STAP
9999 GO TO 9992
```

Programma

```
9900 REM
9901 REM HERNUMMER model 3
9902 LET BVB=100: LET EVB=8999: LET STAP=10: LET NIEUW=100
9903 BORDER 7: PAPER 7: INK 1: CLS
9904 PRINT AT 0,7; INVERSE 1;"REGEL#HERNUMMER#MENU"
9905 PRINT AT 2,17; INVERSE 1;"HUIDIG";AT 2,27; INVERSE 1;'
NIEUW"
9906 PRINT AT 4,0;"Beginregel#="
9907 PRINT AT 4,18+4-LEN (STR$ BVB);BVB
9908 PRINT AT 6,0;"Eindregel#="
```

```
9909 PRINT AT 6,18+4-LEN (STR$ EVB);EVB
9910 PRINT AT 8,0;"Nieuwe#beginregel#-"
9911 PRINT AT 8,18+4-LEN (STR$ NIEUW);NIEUW
9912 PRINT AT 10,0;"Stapgrootte#="
9913 PRINT AT 10,18+4-LEN (STR$ STAP);STAP
9914 PRINT AT 14,0; INVERSE 1;"Tik#ENTER#in#als#er#geen####
####veranderingen#nodig#zijn#######"
9915 REM input parameter
9916 PRINT AT 4,26; FLASH 1; INVERSE 1;">"; FLASH 0; INK 2;
"####"
9917 INPUT S$: IF S$="" THEN  GO TO 9919
9918 LET BVB=INT (VAL S$+.5)
9919 PRINT AT 4,26;"#";AT 4,27+4-LEN (STR$ BVB);BVB
9920 PRINT AT 6,26; FLASH 1; INVERSE 1;">"; FLASH 0; INK 2;
"####"
9921 INPUT S$: IF S$="" THEN  GO TO 9923
9922 LET EVB=INT (VAL S$+.5)
9923 PRINT AT 6,26;"#";AT 6,27+4-LEN (STR$ EVB);EVB
9924 PRINT AT 8,26; FLASH 1; INVERSE 1;">"; FLASH 0; INK 2;
"####"
9925 INPUT S$: IF S$="" THEN  GO TO 9927
9926 LET NIEUW=INT (VAL S$+.5)
9927 PRINT AT 8,26;"#";AT 8,27+4-LEN (STR$ NIEUW);NIEUW
9928 PRINT AT 10,26; FLASH 1; INVERSE 1;">"; FLASH 0; INK 2
;"####"
9929 INPUT S$: IF S$="" THEN  GO TO 9931
9930 LET STAP=INT (VAL S$+.5)
9931 PRINT AT 10,26;"#";AT 10,27+4-LEN (STR$ STAP);STAP
9932 FOR I=1 TO 50: NEXT I
9933 REM zoek beginregel
9934 LET N=23755
9935 IF 256*PEEK N+PEEK (N+1)>=BVB THEN  GO TO 9937
9936 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1): GO TO 9935
9937 CLS : PRINT AT 0,11; FLASH 1; INK 2; INVERSE 1;"HERNUM
MER"
9938 GO SUB 9991
9939 PRINT AT 0,11; INK 2; INVERSE 1;"##OK#KLAAR#"
9940 INPUT """*""#=#STOP,""z""#copie#op#printer##Elke#ander
e#toets#VERVOLG.";S$
9941 IF S$="z" THEN  COPY : GO TO 9940
9942 IF S$<>"*" THEN  GO TO 9903
9943 INPUT """v""#verwijder#het#programma########Elke#ande
re#toets#=#STOP#";k$
9944 IF k$<>"v" THEN  STOP
9945 LET N=23755
9946 IF 256*PEEK N+PEEK (N+1)>=9900 THEN  GO TO 9948
9947 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1): GO TO 9946
9948 LET DA=N+2: LET DL=-4
9949 LET n1=PEEK (N+2)+256*PEEK (N+3): LET DL=DL+n1+4
9950 LET n2=(256*PEEK N+PEEK (N+1)): IF n2<9999 THEN  LET N
=N+3+n1+1: GO TO 9949
9951 LET n1=INT (DL/256): POKE (DA+1),n1: POKE DA,DL-n1*256
: POKE (DA+2),234
9952 PRINT AT 21,0;"Tik#"; FLASH 1;"9900#+<ENTER>"; FLASH 0
;"#in"
```

```
9993 STOP
9991 LET L=NIEUW
9992 IF (256*PEEK N+PEEK (N+1))>EVB THEN  RETURN
9993 POKE N,INT (L/256)
9994 POKE N+1,L-256*INT (L/256)
9995 FOR I=N+4 TO N+3+PEEK (N+2)+256*PEEK (N+3)
9996 IF PEEK I=236 OR PEEK I=237 THEN  PRINT L;"-";CHR$ PEE
K I
9997 NEXT I
9998 LET N=I: LET L=L+STAP
9999 GO TO 9992
```

6.3 Regelblokverwijder

(Copyright Beam Software)

Beschrijving

Dit is ook een programma dat elke Spectrum BASIC-programmeur niet kan missen bij het ontwikkelen van zijn programma's. Als u het hernummerprogramma al bent tegengekomen in dit boek, dan weet u dat dit Regelblokverwijder-programma zeer handig is bij het verwijderen van het hernummerprogramma als dat zijn werk eenmaal gedaan heeft. Daar komt bij dat dit programma ook nog eens zichzelf uitveegt.
Opnieuw worden scherm communicatietechnieken gebruikt om de eerste en de laatste regel van het te verwijderen blok op te vragen. Het programma is 'slim' genoeg om alleen regels groter dan of gelijk aan de eerste regel en kleiner dan of gelijk aan de laatste regel uit te vegen. Dus zelfs als u een niet-bestaand regelnummer opgeeft, dan zal het programma nog geen 'crash' geven. Om het programma te gebruiken moet u de begin- en de eindregel van het te verwijderen blok opgeven. U kunt dus ook maar 1 regel verwijderen door de begin- en eindregel van het blok aan elkaar gelijk te maken. Als het programma klaar is vraagt u het blok te verwijderen door de beginregel in te tikken, gevolgd door ENTER.

Programma-opbouw

Het in dit programma gebruikte algoritme telt alle bytes in het geheugen vanaf het adres wat de regellengte bevat van de beginregel van het blok, tot aan de 'end of line' byte van de eindregel van het blok. Het getelde aantal bytes wordt dan gePOKEd in het adres wat de regellengte van de beginregel bevat en in het daaropvolgende adres wordt 234 (de code voor REM) gePOKEd. In feite wordt hiermee het hele blok in een lang REM-statement gezet. Door nu het nummer van de beginregel in te tikken, gevolgd door ENTER, wordt deze regel in zijn geheel verwijderd.

De opbouw is als volgt:

Invoer begin- en eindregel van te verwijderen blok
Zet HERINDELING boodschap op scherm
Zoek geheugenadres beginregel
Sla adres regellengte op en initialiseer nieuwe lengte
Tel bytes tot einde blok

POKE aantal bytes in adres regellengte
Zet OK KLAAR boodschap op scherm
Zet VERWIJDER PROGRAMMA? boodschap op scherm

Opmerkingen
Als het herindeling-proces klaar is, maar u heeft nog niet het nummer van de beginregel ingetikt, gevolgd door ENTER, dan zijn de regels binnen het blok nog niet verwijderd. Deze kunt u nog steeds aanroepen, met uitzondering van de beginregel want als u die aanroept dan wordt het hele blok alsnog verwijderd.
Mocht u van deze mogelijkheid gebruik willen maken, dan is het wel verstandig om het programma eerst even op cassette weg te zetten.
Dit programma moet u met een MERGE-instructie inladen omdat een LOAD-instructie het programma waarbij u het wilt gebruiken zou uitvegen.

Programma

```
9973 REM
9974 REM REGELBLOK VERWIJDER
9975 OVER 0: FLASH 0: PAPER 7: INK 1: CLS
9976 PRINT AT 0,7; INVERSE 1;"REGELBLOK#VERWIJDER"
9977 PRINT AT 4,0;"VERWIJDER#(INCLUSIEF)"
9978 PRINT AT 6,7;"VANAF#REGEL"
9979 PRINT AT 8,7;"TOT###REGEL";
9980 PRINT AT 6,19; FLASH 1;"("; FLASH 0; INK 2; INVERSE 1;
"####"
9981 INPUT S$: IF VAL S$<0 OR VAL S$>9972 THEN  GO TO 9981
9982 LET SOD=INT (VAL S$)
9983 PRINT AT 6,19;"#####";AT 6,20+4-LEN (STR$ SOD);SOD
9984 PRINT AT 8,19; FLASH 1;")"; FLASH 0; INK 2; INVERSE 1;
"####"
9985 INPUT S$: IF VAL S$<0 OR VAL S$>9972 THEN  GO TO 9985
9986 LET EOD=INT (VAL S$)
9987 PRINT AT 8,19;"#####";AT 8,20+4-LEN (STR$ EOD);EOD
9988 PRINT AT 11,10; FLASH 1; INK 2; INVERSE 1;"HERINDELEN"

9989 LET N=23755
9990 IF 256*PEEK N+PEEK (N+1)>=SOD THEN  GO TO 9992
9991 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1): GO TO 9990
9992 LET DA=N+2: LET DL=-4
9993 LET n1=PEEK (N+2)+256*PEEK (N+3): LET DL=DL+n1+4
9994 LET n2=(256*PEEK N+PEEK (N+1)): IF n2<EOD THEN  LET N=
N+3+n1+1: GO TO 9993
9995 IF n2=EOD THEN  LET DL=DL-n1-4
9996 LET n1=INT (DL/256): POKE (DA+1),n1: POKE DA,DL-n1*256
: POKE (DA+2),234
9997 PRINT AT 11,10;"############";AT 11,12; INK 2; INVERSE
1;"OK#KLAAR"
9998 PRINT AT 15,0;"TIK#"; FLASH 1;SOD;"#+<ENTER "; FLASH 0
;"#IN#OM#BLOK#TE#####VERWIJDEREN"
9999 STOP
```

6.4 Machinetaalmonitor

(Copyright Beam Software)

Op sommige plaatsen in dit boek wordt gesproken over machinetaal-programma's die efficiënter zijn dan BASIC, en in sommige gevallen zijn dergelijke programma's gebruikt om te laten zien dat ze sneller zijn en geheugenruimte besparen.
U zult zich ondertussen dan ook wel afvragen wat nu precies het verschil is tussen BASIC en machinetaal.

Om te beginnen moet u eerst weten dat de Spectrum eigenlijk uw BASIC programma nooit uitvoert. De chip die in deze computer al het werk doet, de Z80 begrijpt helemaal niets van BASIC. Het enige wat die chip begrijpt is een zeer beperkt aantal eigen instructies, zoals bijvoorbeeld tel twee getallen bij elkaar op, trek twee getallen van elkaar af, vergelijk twee getallen met elkaar enz.
Nu is het wel zo dat met deze simpele instructies bijna alles gedaan kan worden. De meest ingewikkelde bewerking is namelijk altijd te herleiden tot een reusachtige hoeveelheid zeer eenvoudige problemen.
Maar zelfs een relatief simpele instructie zoals vermenigvuldigen wordt door de Z80 niet begrepen. Om dergelijke taken te verrichten moet er een programma geschreven worden.

Uw Spectrum werd geleverd met een zeer lang machinetaal-programma kant en klaar ingebouwd. Dit programma, dat 16K bestrijkt, zit in een chip gebakken zodat u het niet elke keer hoeft in te laden als u de computer wilt gebruiken.
Deze chip wordt de ROM-chip genoemd naar het Engelse Read Only Memory, wat dus betekent dat u de informatie die hier in staat niet kunt veranderen (dit in tegenstelling tot zgn. RAM-chips 'Random Access Memory' die u kunt uitlezen en kunt veranderen).
Dit programma in de ROM leest wat u de computer via het toetsenbord vertelt en zorgt ervoor dat het gewenste resultaat geleverd wordt. Een eenvoudige regel in BASIC kan honderden regels machinetaal nodig hebben om zijn taak uit te voeren.
U zult ondertussen wel begrijpen dat het voor ons mensen veel gemakkelijker is om in BASIC te programmeren, maar dat het programma niet altijd zo snel zal lopen als wij ons dat wensen.

'Machinetaalmonitor' pretendeert geenszins u te leren programmeren in machinetaal. Hele boeken zijn hier over vol geschreven en als u geïnteresseerd bent kunnen we u wel twee boeken aanraden:

* Machinetaal voor de ZX Spectrum door William Tang
  (uitgave van Kluwer Technische Boeken)
* Understanding Your Spectrum door Dr. Ian Logan.

De opbouw van machinetaal-programma's

De opbouw van machinetaal verschilt nogal van BASIC. Enkele belangrijke verschillen zijn:

* Variabelen bestaan niet.
  Er bestaat geen direct equivalent voor bijv.
  LET score = 3

Wat de Z80 betreft bestaat er alleen maar zo iets als een geheugenadres.
Als u de Spectrum-handleiding bekijkt dat vindt u op blz. 173 - 176 een lijst geheugenadressen die door de ROM als variabelen gebruikt worden.
De waarden in deze adressen kunt u veranderen, met uiteenlopende resultaten, maar meer overeenkomsten met variabelen bestaan er eigenlijk niet.

* Regelnummers bestaan niet
Als de Z80 instructies uitvoert dan gaat dat één voor een, in de volgorde waarin hij ze tegenkomt.
Er bestaan wel instructies zoals GOTO en GOSUB, maar deze instructies springen naar een opgegeven geheugenadres. Dat betekent dat in het algemeen machinetaalprogramma's ontworpen zijn om te functioneren op een bepaald gebied in het geheugen.
Er zijn wel programma's die niet plaatsgebonden zijn, maar die worden later behandeld.

* Alle gebruikte getallen liggen ~~tussen de~~ van de 0 - 255 of 0 - 65535 (bij gebruik van respectievelijk een of twee geheugenplaatsen). Het getal dat dus per geheugenadres kan worden gebruikt ligt dus tussen 0 en 255. Om deze waarden gemakkelijk te kunnen gebruiken is er een aparte notatie ontwikkeld, de zgn. hexadecimale notatie (afgekort tot Hex of H). Het gaat te ver om dat hier uit te leggen, maar achterin de Spectrum-handleiding (blz. 183-188) staat een omrekentabel.

Wat is PEEK en POKE?
Nu u een beetje beter begrijpt hoe de Spectrum werkt snapt u wel dat er geen eenvoudig equivalent is voor LET a = b.
Wat er het dichtste bijkomt is om aan a en b te denken in termen van de inhoud van bepaalde geheugenadressen in plaats van als variabelen. Hoe weet u nu de inhoud van een adres?
Met de BASIC-opdracht PEEK kunt u de inhoud van geheugenplaatsen uitlezen zonder dat u die inhoud verandert. Zo komt u te weten hoe groot b is. Denk er wel aan dat een adres altijd een waarde tussen 0 - 255 zal bevatten.

U kunt de inhoud van een adres veranderen met een POKE-opdracht. Met deze niet bepaald subtiele en vrij definitieve opdracht kunt u behoorlijk wat schade aanrichten in uw programma's (niet in de computer). POKE maar eens wat willekeurige getallen in de systeem variabelen (blz. 173-176 van de Spectrum-handleiding), vooral op die plaatsen waar een X bijstaat.

Hex-notatie
Als u op blz. 183-188 van de Spectrum-handleiding kijkt dan treft u daar een kolom aan met Hex-getallen.
Wat opvalt is dat:

Elk getal is altijd twee karakters lang.
Alleen de cijfers 0-9 en de letters A-F worden gebruikt.

Dit is het grote voordeel van Hex-notatie

De letters A-F staan voor de waarden 10-15. Om nu een Hex-notatie om te rekenen in decimale notatie moet u de waarde in de eerste kolom vermenigvuldigen met 16 en daar de waarde uit de tweede kolom bij optellen (hetzelfde als bij een decimale notatie maar die werkt met een basis van 10 in plaats van 16)

Bijv.: OA Hex = 10 decimaal
11 Hex = 17 decimaal

Het Machinetaal-Monitorprogramma

Hiermee kunt u ieder adres bekijken in de Spectrum, of dat nu in ROM is, of in RAM (het normale geheugen bestaande uit scherm, systeem variabelen, BASIC-programma, BASIC-variabelen, leeg geheugen en vrij programmeerbare karakters, (zie schema blz. 165 van de handleiding). De inhoud van RAM-adressen kunt u met dit programma naar wens veranderen

Als u erg veel waarden in een bepaald geheugengebied moet invoeren dan is dit zeker niet het beste programma om mee te werken (zie hiervoor andere programma's in dit boek, bijv. het DAM-programma), dit programma is uitstekend geschikt om een geheugengebied te onderzoeken en aan te passen.

Elke regel op het scherm toont de inhoud van 10 adressen in Hex-notatie. Gebruik de pijltoetsen (CAPS SHIFT 5-8) om de cursor te besturen. Door een andere toets dan een pijltoets te gebruiken kan de inhoud van het adres onder de cursor veranderd worden. Denk er wel aan dat u de gewenste waarde in Hex-notatie moet invoeren.

Interessante oefeningen

* Probeer de inhoud van een geheugenadres in de ROM te veranderen (adres < 16384).
* Probeer de inhoud van een adres tussen 22528 en 23295 te veranderen. Dit is het gebied van de schermattributen
* Probeer de inhoud van een adres tussen 16384 en 22537 te veranderen. Dit is het gebied van het scherm zelf.
  Zoals u misschien wel weet beslaat elk karakter op het scherm acht geheugenplaatsen die echter niet opvolgend zijn. Probeer het verband tussen deze acht te vinden.

Met deze oefeningen kunt u geen schade toebrengen aan de programma's of de computer. Schade toebrengen aan de computer door programmeren, bijv. een POKE-opdracht is zonder meer onmogelijk. Het ergste wat er kan gebeuren is dat de computer niet meer luistert naar het toetsenbord. In dat geval moet u even de stroom uitschakelen en bent u alles wat er in stond kwijt.

Verplaatsbare Machinetaalprogramma's

Sommige machinetaalprogramma's worden verplaatsbaar genoemd (relocatable), wat betekent dat ze overal in het geheugen kunnen functioneren.

Zulke programma's zijn uitstekend geschikt als hulproutine bij een BASIC-programma omdat ze overal waar ruimte is geplaatst kunnen worden.
De schermschuif-routine uit het KIDNAP-programma in dit boek is daar een goed voorbeeld van.

Hier volgt nog zo'n nuttig verplaatsbaar programma, een regelhernummer-routine. Als u dit programma vergelijkt met de BASIC-versie elders in het boek dan merkt u dat dit machinetaal-programma veel korter is, slechts 27 bytes lang, en stukken sneller werkt.
Om dit programma in het geheugen te krijgen kunt u het korte BASIC-programma gebruiken of u kunt de bytes met behulp van uw monitorprogramma in Hex-notatie invoeren vanaf adres 32500 (zie eerste kolom 'assembly-uitdraai' verderop).
Nadat het programma is ingevoerd, is het verstandig om het eerst weg te schrijven op cassette met behulp van een
SAVE "" CODE-opdracht.
Om een machinetaalprogramma op te starten gebruikt u een USR instructie waarin het beginadres van het programma specificeert, bijv. PRINT USR 32500 of LET L = USR 32500.
Zorg er wel voor dat de machinetaal in een veilig gebied staat, bijv. boven een CLEAR of in het VPK-gebied.
Dit machinetaalprogramma hernummert alle regels vanaf regel 100 in stappen van 10. Deze waarden kunt u veranderen door het vijfde getal (nu Hex 5A=90: 100 10) en het dertiende getal (de stap, nu Hex 0A=10) aan te passen aan uw wensen. Het programma hernummert geen GOTO en GOSUB-opdrachten.

'Assembly-uitdraai' machinetaalhernummerprogramma:

```
11 CA 5C    LD DE, 5CCAH    ;begin BASIC
21 5A 00    LD HL, 90       ;eerste regelstap
13 volgr    INC DE
1A          LD A, (DE)      ;wat staat daar?
FE 28       CP 28H          ;code einde regel?
D0          RET NC          ;nee => klaar!
01 0A 00    LD BC, 10       ;stapgrootte
09          ADD HL,BC       ;nieuw regelno
EB          EX DE, HL       ;tijdelijke ruil
72          LD (HL),D       ;sla nieuw regno op
23          INC HL
73          LD (HL),E
23          INC HL          ;zoek regellengte
4E          LD C, (HL)      ;sla op in BC
23          INC HL
46          LD B, (HL)
09          ADD HL, BC      ;pos einde regel
EB          EX DE, HL       ;einde ruil
18 EB       JR volgr        ;ga naar volgende
                            ;regel
```

Programma

```
  10 REM monitor
 100 PRINT  INK 7; PAPER 2;AT 0,0;'MACHINE#TAAL#MONITOR####
###"
 110 INPUT "Start#adres##(decimaal)#";s
 120 LET s=10*INT (s/10)
 130 LET x=0: LET y=0
 140 GO SUB 1000
 150 PRINT  INK 7; PAPER 2;AT 0,0;"MACHINE#TAAL#MONITOR####
###"
 160 PRINT  OVER 1; FLASH 1;AT 3+2*x,3+3*y;'##"
 170 LET a$=INKEY$
 180 IF CODE a$<12 AND CODE a$>7 THEN  GO TO 600
 190 IF CODE a$=0 THEN  GO TO 170
 200 INPUT "VERANDER#BYTE#WAARDE#IN#(Hex)#";a$
 210 IF LEN a$<>2 THEN  GO TO 160
 220 GO SUB 500
 230 IF v<0 OR v>15 THEN  GO TO 170
 240 LET h=v
 250 LET a$=a$(2)
 260 GO SUB 500
 270 IF v<0 OR v>15 THEN  GO TO 170
 280 LET l=v
 290 LET v=16*h+l
 300 POKE s+10*x+y,v
 310 PRINT  OVER 1; FLASH 0;AT 3+2*x,3+3*y;"##"
 320 IF y=9 THEN  GO SUB 1000
 330 LET y=y+1
 340 IF y>9 THEN  LET y=0: LET x=x+1
 350 IF x<10 THEN  GO TO 140
 360 GO TO 730
 500 LET v=CODE a$-48-7*(CODE a$>64)-32*(CODE a$>96)
 510 RETURN
 600 PRINT  OVER 1; FLASH 0;AT 3+2*x,3+3*y;'##'
 610 GO TO 540+10*CODE a$
 620 LET y=y-1: GO TO 660
 630 LET y=y+1: GO TO 700
 640 LET x=x+1: GO TO 720
 650 LET x=x-1: GO TO 680
 660 IF y=0 THEN  GO TO 160
 670 LET y=9: LET x=x-1
 680 IF x=0 THEN  GO TO 140
 690 LET s=s-10: CLS : GO TO 130
 700 IF y<10 THEN  GO TO 160
 710 LET y=0: LET x=x+1
 720 IF x<10 THEN  GO TO 140
 730 IF x=10 THEN  CLS : LET s=s+10*x: GO TO 130
 740 GO TO 140
1000 IF s<0 OR s>65530 THEN  GO TO 110
1010 PRINT AT 2+2*x,0;s+10*x: PRINT TAB 2;
1020 FOR i=s+10*x TO s+10*x+9
1030 IF i>65535 THEN  LET i=s+10*1+9: GO TO 150
1040 LET v=PEEK i: LET h=INT (v/16): LET l=v-16*h
1050 PRINT "#"+CHR$ (h+48+7*(h>9))+CHR$ (l+48+7*(l>9));
1060 NEXT i
1070 RETURN
```

Programma

```
 100 REM machinetaal hernummer
 105 CLEAR 32500: LET a=32500
 110 READ n: POKE a,n
 120 LET a=a+1: GO TO 110
 130 DATA 17,202,92,33,90,0,19,26,254,40,208,1,10,0,9,235,1
14,35,115,35,78,35,70,9,235,24,235
 140 FOR i=0 TO 26
 150 PRINT PEEK (32500+i);"#";
 160 NEXT i
```

7. Zakelijke programma's

7.1 Loonlijst
(Copyright Beam Software)

Dit programma berekent de inkomens van werknemers en zet een overzicht op het scherm dat gebruikt kan worden om loonstaten te produceren. Het overzicht begint met de optie om

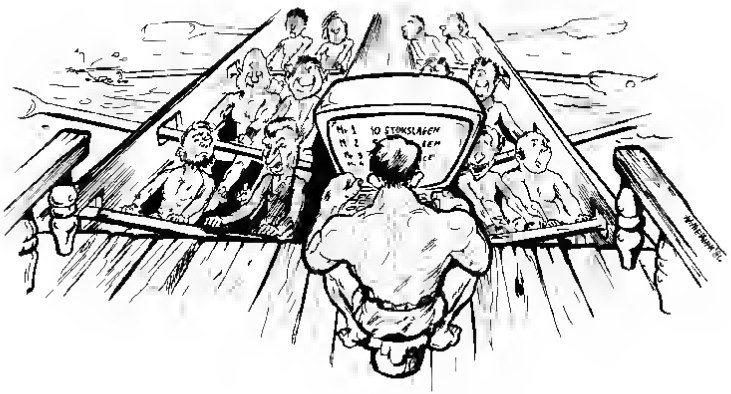
ot Invoer naam van werknemer, uurloon en overwerkhonorering
Indien gewenst kunnen deze werknemergegevens op cassette worden gezet

ot Inladen van eerder op cassette gezette werknemergegevens

Aan het eind van elke loonperiode worden de werknemersuren ingevoerd (normale uren en overwerk) waarna het programma het brutosalaris voor elke werknemer berekent. Hoewel dit programma maar geschikt is voor maximaal 18 werknemers kan er bij gebruik in een kleine onderneming toch veel rekenwerk bespaard worden.
Alle invoer en uitvoer wordt via het scherm geregeld, een knipperende cursor op het scherm geeft aan met welk onderdeel er gewerkt wordt. Alle gegevens worden door de computer dusdanig gehanteerd dat ze netjes onder elkaar komen te staan.
Nadat alle invoer en berekeningen afgesloten zijn kan met elke toets een van de volgende drie hoofdschermen worden opgeroepen.

Werknemergegevens
Uurloonoverzicht
Salarisoverzicht

De salarisberekening is tot op de cent nauwkeurig



Programma-opbouw

Dit programma is één van de drie Spectrum-financiële modellen in dit boek. Een van de grote verschillen tussen dit soort programma's en bijv. spelletjes is dat alle invoer en uitvoer op een zeer gebruikersvriendelijke manier geregeld moet worden. Wat het programmeren zelf betreft zult u merken dat dit voornamelijk bestaat uit berekeningen en 'string'-handelingen.

```
Initialisatie
  initialiseer vlaggen drie schermen
  initialiseer schermvariabelen

Werknemergegevens
  als 'I' of 'T' (invoer nieuwe gegevens)
    invoer aantal werknemers (1-18)
    dimensioneer de volgende arrays
      E$ werknemergegevens
      R$ normale uren
      O$ overwerk
      G$ brutosalaris
  als 'L' of 'l' (laadt gegevens op van cassette)
    invoer werknemergegevens van cassette
    dimensioneer bovenstaande arrays behalve E$
: zet werknemergegevens op scherm
  als RM-vlag is invoer (RM=0)
    zet RM op 1 (= scherm)
    invoer NAAM werknemer en uurloon
    invoer overwerkhonorering
    sla gegevens op
    return
  zet gegevens opnieuw op scherm
    optiegegevens op printer
    return

Loonlijst uurloonoverzicht
  zet uurloonoverzicht op scherm
  als PM-vlag is invoer (PM=0)
    zet PM-vlag op 1 (scherm)
    invoer normale uren en overwerk
    return
  zet uurloonoverzicht opnieuw op scherm
    optie-overzicht op printer
  return

Loonlijst salarisoverzicht
  als PR-vlag is bereken (PR=0)
    zet PR-vlag op 1 (scherm)
    initialiseer TP (totaal brutosalaris)
    bereken en tel op alle salarissen
```

```
    zet salarissen in corresponderende arrays
  zet salarisoverzicht op scherm
    optie-overzicht op printer
    return

Hoofdlus
  Werknemergegevens
L1 Loonlijst uurloonoverzicht
  Loonlijst salarisoverzicht
  GOSUB I
  GOTO L1
```

Opmerkingen

De werknemergegevens-array is als volgt opgebouwd:

* eerste item
  plaats 1-5 aantal werknemers in array
  plaats 16-21 overwerkhonorering

* volgende item
  plaats 1-15 naam werknemer
  plaats 16-21 uurloon werknemer

Er zijn drie hoofd-arrays die ieder voornamelijk twee taken moeten uitvoeren; invoer/berekening en overzicht op scherm.
Om te bepalen welke van deze twee taken uitgevoerd moet worden zet het programma in het begin de drie variabelen die als vlag dienst doen op invoer/berekening d.w.z. 0, om ze nadat een array de invoer/berekening routine doorlopen heeft op het schermoverzicht te zetten d.w.z. 1.
Dit betekent dat elke keer als er daarna van een array gebruik gemaakt wordt deze array op het scherm zal verschijnen.
De routine op regel 3000 zorgt ervoor dat elke string bestaande uit een getal, waarvan het aantal cijfers voor de komma bekend is, wordt omgezet in een string met hetzelfde aantal cijfers voor de komma en twee cijfers na de komma. Dit geeft de mogelijkheid om bijv. alleen maar '34' te hoeven invoeren in plaats van '34.00'.

Vergeet bij het invoeren van getallen niet dat de Spectrum van Engelse origine is; niet gehele getallen moeten worden ingevoerd als bijv. '34.50' in plaats van '34,50'.
Dit geldt trouwens voor vrijwel alle computersystemen.

Programma

```
 100 REM LOONLIJST
 120 LET rm=0: LET pm=rm: LET pr=rm: REM zet vlaggen
 130 OVER 0: PAPER 7: INK 1: FLASH 0: CLS : GO TO 9000
1000 REM
1010 REM initialiseer routine
```

```
1020 CLS : PRINT AT 0,6; INVERSE 1; INK 1;"WERKNEMER#GEGEVE
NS"
1030 PRINT AT 6,1; INK 1;"INVOER#nieuwe#gegevens#----#I"
1040 PRINT AT 8,1; INK 1;"LAAD[####oude#gegevens#----#L"
1050 PRINT AT 15,8; INK 1;"Toets#'"; FLASH 1;"I"; FLASH 0;
"'#of#'"; FLASH 1;"L"; FLASH 0;"'"
1060 LET k$=INKEY$: IF k$<>"I" AND k$<>"L" AND k$<>"1" AND
k$<>"1" THEN  GO TO 1060
1070 IF k$="1" OR k$="1" THEN  LET k$=CHR$ (CODE k$-32)
1080 PRINT AT 15,15; INK 1; INVERSE 1;"I";AT 15,22;"L": PRI
NT AT 18,12;
1090 IF k$<>"L" THEN  GO TO 1150
1100 PRINT  INVERSE 1; INK 2;"OPLADEN";: LET rm=1
1110 INPUT "Tik#<spatie><ENTER>#in#";k$
1120 IF k$<>"#" THEN  GO TO 1110
1130 LOAD "loon" DATA E$()
1140 LET NE=VAL E$(1,1 TO 2): LET P=VAL E$(1,15 TO 21): GO
TO 1190
1150 PRINT  INVERSE 1; INK 2;"#INVOER#": PRINT AT 21,0; INK
 1; INVERSE 1; FLASH 1;"#"; FLASH 0; INVERSE 0;"aantal#werk
nemers#(1#-#18):#"; INVERSE 1;"##"
1160 INPUT NE: IF NE<1 OR NE>18 THEN  GO TO 1160
1170 PRINT AT 21,30;"##";CHR$ 8;CHR$ 8; INK 1; INVERSE 1;NE

1180 DIM E$(NE+1,21)
1190 DIM R$(NE,6)
1200 DIM O$(NE,6)
1210 DIM G$(NE,8)
1220 FOR I=1 TO 75: NEXT I
1230 INK 1: PAPER 7: CLS : PRINT AT 0,8; INVERSE 1;"WERKNEM
ER#GEGEVENS"
1240 PRINT AT 2,2; INVERSE 1;"NAAM#WERKNEMER";AT 2,21;"UURL
OON"
1250 IF rm=1 THEN  GO TO 1500
1260 REM invoer werknemer gegevens
1270 LET rm=1
1280 FOR I=2 TO NE+1
1290 LET E$(I,1 TO 15)="#": LET E$(I,16 TO 21)="000000"
1300 LET In=I+3-2
1310 PRINT AT In,1;: IF I>10 THEN  PRINT AT In,0;
1320 PRINT I-1; FLASH 1; INVERSE 1;" ";AT In,3; FLASH 0; IN
K 2;"################"
1330 INPUT S$: IF S$="*" THEN  RETURN
1340 LET E$(I,1 TO 15)=S$: PRINT AT In,2;"#";E$(I,1 TO 15)
1350 PRINT AT In,20; INVERSE 1; FLASH 1;">";AT In,22; FLASH
 0; INK 2;"######"
1360 INPUT S$: IF VAL S$<0 THEN  GO TO 1360
1370 LET ns=3: GO SUB 3000: REM invoer numeieke string
1380 PRINT AT In,20;"#";AT In,22;"###### ";AT In,22;S$: LET
E$(I,16 TO 21)=S$
1390 NEXT I
1400 PRINT AT 21,6; INVERSE 1;"Overwerkhonorering";AT 21,25
; FLASH 1;"#"; FLASH 0; INK 2;AT 21,27;"#####"
1410 INPUT S$: IF VAL S$<0 THEN  GO TO 1410
```

```
1420 LET F=VAL S$: PRINT AT 21,25;"#######";AT 21,25;F: LET
 E$(1,16 TO 21)=S$
1430 LET E$(1,1 TO 2)=STR$ NE
1440 FOR I=1 TO 50: NEXT I
1450 INPUT "GEGEVENS#op#cassette#zetten?#";k$
1460 IF k$="" THEN  GO TO 1450
1470 IF k$(1)="j" OR k$="J" THEN  SAVE "loon" DATA E$(): RE
TURN
1480 IF k$(1)<>"n" AND k$(1)<>"N" THEN  GO TO 1450
1490 RETURN
1500 REM zet werknemergegevens opnieuw op scherm
1510 LET NE=VAL (E$(1,1 TO 2)): LET F=VAL  (E$(1,16 TO 21))
1520 FOR I=2 TO NE+1
1530 LET ln=I+3-2
1540 PRINT AT ln,2;E$(I,1 TO 15);AT ln,23;E$(I,16 TO 21)
1550 NEXT I
1560 PRINT AT 21,6; INVERSE 1;"OVERWERKHONORERING"
1570 PRINT AT 21,25;F
1580 LET k$=INKEY$: IF k$="" THEN  GO TO 1580
1590 IF k$="z" THEN  COPY : GO TO 1580
1600 RETURN
2000 REM
2010 REM invoer/scherm uurloon overzicht
2020 CLS : PRINT AT 0,8; INVERSE 1;"UURLOON#OVERZICHT"
2030 PRINT AT 1,17; INVERSE 1;"-----#UREN#-----"
2040 PRINT AT 2,0; INVERSE 1;"WERKNEMER";AT 2,19;"NORM";AT
2,27;"OVW"
2050 IF pm=1 THEN  GO TO 2500
2060 LET pm=1: LET TR=0: LET TO=TR
2070 FOR I=2 TO NE+1
2080 LET ln=I+3-2
2090 PRINT AT ln,0;E$(I,1 TO 15)
2100 PRINT AT ln,16; FLASH 1;""; FLASH 0; INVERSE 1; INK 2
;"######"
2110 INPUT n1: IF n1<0 THEN  GO TO 2110
2120 LET S$=STR$ n1: LET ns=3: GO SUB 3000
2130 PRINT AT ln,16;"#";S$: LET TR=TR+n1: LET R$(I-1)=S$
2140 PRINT AT ln,24; FLASH 1;">"; FLASH 0; INK 2; INVERSE
 1;"######"
2150 INPUT n1: IF n1<0 THEN  GO TO 2150
2160 LET S$=STR$ n1: LET ns=3: GO SUB 3000
2170 PRINT AT ln,24;"#";S$
2180 LET TO=TO+n1: LET O$(I-1)=S$
2190 NEXT I
2200 LET S$=STR$ TR: LET ns=4: GO SUB 3000
2210 PRINT AT 21,8; INVERSE 1;"TOTAAL";AT 21,16; INVERSE 0;
S$
2220 LET S$=STR$ TO: GO SUB 3000
2230 PRINT AT 21,24;S$
2240 RETURN
2500 REM zet uurloon overzicht  op scherm
2510 FOR I=1 TO NE
2520 LET ln=I+3-1
2530 PRINT AT ln,0;E$(I+1,1 TO 15);AT ln,17;R$(I,1 TO 6);AT
 ln,25;O$(I,1 TO 6)
```

```
2540 NEXT I
2550 PRINT AT 21,8; INVERSE 1;"TOTAAL"
2560 LET S$=STR$ TR: LET ns=4: GO SUB 3000
2570 PRINT AT 21,16;S$
2580 LET S$=STR$ TO: GO SUB 3000
2590 PRINT AT 21,24;S$
2600 LET k$=INKEY$: IF k$="" THEN  GO TO 2600
2610 IF k$="z" THEN  COPY : GO TO 2600
2620 RETURN
3000 REM
3010 REM string invoer waarden
3020 LET T$=STR$ (INT VAL S$)
3030 FOR J=1 TO (ns-LEN T$): LET T$="#"+T$: NEXT J
3040 LET U$=STR$ (INT ((VAL S$-VAL T$)*100+.5)): LET T$=T$+
"."
3050 IF VAL U$=0 THEN  GO TO 3090
3060 FOR J=1 TO LEN U$: IF U$(J TO J)="." THEN  GO TO 3080
3070 NEXT J
3080 LET U$=U$(1 TO (J-1))
3090 IF VAL U$<10 THEN  LET U$="0"+U$
3100 LET S$=T$+U$
3110 RETURN
4000 REM
4010 REM bereken/scherm salaris overzicht
4020 IF pr=1 THEN  GO TO 4100
4030 LET pr=1: LET TP=0
4040 FOR I=1 TO NE
4050 LET n1=(VAL R$(I)+VAL O$(I)*F)*(VAL E$(I+1,16 TO 21))
4060 LET S$=STR$ n1: LET ns=5: GO SUB 3000
4070 LET G$(I,1 TO 8)=S$
4080 LET TP=TP+n1
4090 NEXT I
4100 CLS : PRINT AT 0,8; INVERSE 1;"SALARIS#OVERZICHT"
4110 PRINT AT 2,0; INVERSE 1;"WERKNEMER";AT 2,21;"BRUTO#LOO
N";AT 3,19;"F1"
4120 FOR I=1 TO NE
4130 LET ln=I+3-1
4140 PRINT AT ln,0;E$(I+1,1 TO 15);AT ln,22;G$(I,1 TO 8)
4150 NEXT I
4160 PRINT AT 21,6; INVERSE 1;"TOTAAL#LOON";AT 21,19;"F1";
INVERSE 0;
4170 LET S$=STR$ TP: LET ns=6: GO SUB 3000
4180 PRINT "#";S$
4190 LET k$=INKEY$: IF k$="" THEN  GO TO 4190
4200 IF k$="z" THEN  COPY : GO TO 4190
4210 RETURN
9000 REM
9010 REM hoofd loop
9020 GO SUB 1000: REM initialiseer
9030 GO SUB 2000: REM uurloon overzicht
9040 GO SUB 4000:  REM salaris overzicht
9050 GO SUB 1230: REM gegevens
9060 GO TO 9030
```

7.2 Verkoopanalyse
(Copyright Beam Software)

Dit programma gebruikt de verkoopcijfers over een aantal jaren van een onderneming om te bepalen wat de seizoenschommelingen en de algemene trend zijn geweest over deze jaren
Het programma is zeer nuttig voor elke onderneming die te maken heeft met sterk seizoengebonden verkopen. In een boekenuitgeverij is het bijvoorbeeld normaal dat zo'n 25% van de jaarlijkse omzet in de maanden november en december binnengehaald wordt.
De informatie die het programma nodig heeft zijn de verkoopcijfers per kwartaal over een zo groot mogelijk aantal jaren (max 18).
Hierna berekent het programma de seizoenaandelen door de werkelijke verkopen te delen door de gemiddelde kwartaalverkopen over alle jaren
Het gemiddelde van elk seizoenaandeel over alle jaren geeft de seizoenindex.
Om een snelle visuele toetsing van de groei van de onderneming mogelijk te maken zet het programma de jaarlijkse verkoopcijfers van het bedrijf in grafiekvorm op het scherm

Opbouw
Net als bij de andere zakelijke programma's in dit boek is hier veel aandacht besteed aan de wisselwerking tussen gebruiker en computer. Dit soort programma's moet namelijk ook gebruikt kunnen worden door mensen die geen verstand hebben van computers
De invoer en uitvoer van gegevens loopt daarom ook via het scherm. Het grote voordeel hiervan is dat de gebruiker meteen ziet hoeveel en welke informatie hij ingevoerd heeft.
De resultaten verschijnen op twee verschillende schermen, een voor de werkelijke verkoopcijfers en een voor seizoenindexen. Door het indrukken van welke toets dan ook kunnen deze twee schermen gewisseld worden. Tevens is er de optie om de resultaten naar de printer te sturen
De grafiek maakt gebruik van de PLOT en DRAW instructies van de Spectrum. Door bij het opzetten van de grafiek-assen de schaal aan te passen aan de minimum- en maximumomzet over de jaren, benut de grafiek het maximaal mogelijke schermoppervlak.

Programma

```
  100 REM
  110 REM SEIZOENS INDEX
  120 BORDER 7: PAPER 7: INK 1: FLASH 0: OVER 0: CLS
  130 GO TO 8000
 1000 REM
 1010 REM INPUT AANTAL JAREN
 1020 PRINT AT 0,9; INVERSE 1;"SEIZOENS#-#INDEX"
 1030 PRINT AT 4,0;"OMZET#VERLOOP#(inclusief)"
 1040 PRINT AT 7,6;"VANAF#JAAR"
 1050 PRINT AT 10,6;"TOT#JAAR"
 1060 PRINT AT 7,17; FLASH 1; INVERSE 1;" "; FLASH 0; INK 2;
 INVERSE 1;"####"
 1070 INPUT S$: LET y1=INT (VAL S$): IF y1<1900 OR LEN (STR$
 y1)<4 THEN  GO TO 1070
 1080 PRINT AT 7,17;"#";y1
```

```
1090 PRINT AT 10,17; FLASH 1; INVERSE 1;">"; FLASH 0; INK 2
; INVERSE 1;"####"
1100 INPUT S$: LET y2=INT (VAL S$): IF y2<1900 OR LEN (STR$
 y2)>4 OR y2<y1 OR y2>y1+17 THEN  GO TO 1100
1110 PRINT AT 10,17;"#";y2
1120 LET NY=y2-y1+1
1130 DIM Y$(NY,4): REM jaar
1140 DIM H$(NY,4,4): REM omzet verloop
1150 DIM A$(NY,7): REM jaarlijksw gem. omzet
1160 DIM R$(NY,4,6): REM aandeel
1170 DIM D$(4,6): REM seizoens-index
1172 DIM D(4): REM som kwartaal ratios
1174 DIM C(NY): REM teken boog dy
1180 DIM C(NY): REM teken boog dx
1190 DIM C(NY): REM teken boog dy
1200 FOR I=y1 TO y2: LET Y$(I-y1+1)=STR$ I: NEXT I
1210 RETURN
2000 REM
2010 REM invoer omzet verloop
2020 CLS : PRINT AT 0,10; INVERSE 1;"OMZET#VERLOOP"
2030 PRINT AT 1,26; INVERSE 1;"GEM.";AT 2,0; INVERSE 1;"JAA
R";AT 2,5; INVERSE 1;"KWT1";AT 2,10;"KWT2";AT 2,15; INVERSE
 1;"KWT3";AT 2,20; INVERSE 1;"KWT4";AT 2,26; INVERSE 1;"OMZ
ET"
2040 IF hm=1 THEN  GO TO 2500
2050 FOR I=1 TO NY
2060 LET ln=I+3-1: PRINT AT ln,0;Y$(I)
2070 LET hp=4: LET ts=0
2080 FOR J=1 TO 4
2090 PRINT AT ln,hp; FLASH 1;">"; INVERSE 1; FLASH 0; INK 2
;"####"
2100 INPUT S$: IF VAL S$<0 THEN  GO TO 2100
2110 LET n1=INT (VAL S$+.5): LET ts=ts+n1: LET S$=STR$ n1
2120 LET Z$="####": LET Z$(5-LEN S$ TO )=S$
2130 PRINT AT ln,hp;"#";Z$
2140 LET H$(I,J)=Z$: LET hp=hp+5
2150 NEXT J
2160 LET S$=STR$ (ts/4): LET n=4: LET nd=2
2170 GO SUB 7000
2180 PRINT AT ln,hp+1;S$
2190 LET A$(1)=S$
2200 NEXT I
2210 LET hm=1: GO TO 2550
2500 REM
2510 REM scherm
2520 FOR I=1 TO NY
2530 PRINT AT I+3-1,0;Y$(I);"#";H$(I,1);"#";H$(I,2);"#";H$(
1,3);"#";H$(I,4);"#";A$(1)
2540 NEXT I
2550 INPUT "Copie#op#printer#(j/n)#?#";k$
2560 IF k$="j" OR k$="J" THEN  COPY : GO TO 2550
2570 IF k$="N" OR k$="n" THEN  RETURN
2580 IF k$="E" THEN  CLEAR : STOP
2590 GO TO 2550
3000 REM
3010 REM bereken seizoensaandelen
```

```
3020 CLS : PRINT AT 0,3; INVERSE 1;"BEREKEND#SEIZOENS-AANDE
EL"
3030 PRINT AT 2,0; INVERSE 1;"JAAR";AT 2,6;"KWT1";AT 2,13;'
KWT2";AT 2,20;"KWT3";AT 2,27;"KWT4"
3040 PRINT AT 21,0; INVERSE 1;"INDEX"
3050 IF cm=1 THEN  GO TO 3500
3060 PRINT AT 1,11; FLASH 1; INK 3;"EVEN#GEDULD!"
3070 FOR I=1 TO 4: LET Q(I)=0: NEXT I
3080 FOR I=1 TO NY
3090 LET n1=VAL  A$(I)
3100 FOR J=1 TO 4
3110 LET n2=VAL H$(I,J): LET n3=n2/n1
3120 LET S$=STR$ n3: LET ns=1: LET nd=4
3130 GO SUB 7000
3140 LET R$(I,J)=S$: LET Q(J)=Q(J)+VAL S$
3150 BEEP .03,12
3160 NEXT J
3170 NEXT I
3180 FOR I=1 TO 4
3190 LET n1=Q(I)/NY
3200 LET S$=STR$ n1: LET ns=1: LET nd=4
3210 GO SUB 7000
3220 LET D$(I)=S$
3230 BEEP .05,24
3240 NEXT I
3250 PRINT AT 1,11;"#"; INVERSE 1; INK 3;"OK#KLAAR"; INVERS
E 0; INK 1;"###"
3260 LET cm=1
3270 NEXT I
3280 PRINT AT 1,11;"#"; INVERSE 1; INK 3;"OK#KLAAR"; INVERS
E 0; INK 1;"##"
3290 LET cm=1
3500 REM
3510 REM zet aandeel/index scherm op
3520 FOR I=1 TO NY
3530 PRINT AT I+3-1,0;Y$(1);"#";R$(I,1);'#";R$(I,2);"#";R$(
I,3);"#";R$(I,4)
3540 NEXT I
3560 PRINT AT 21,5;D$(1);"#";D$(2);"#";D$(3);'#";D$(4)
3570 INPUT "Copie#op#printer#(j/n)#?#";I$
3580 IF I$="j" OR I$="J" THEN  COPY : GO TO 3570
3590 IF I$="n" OR I$="N" THEN  RETURN
3600 IF I$="E" THEN  CLEAR : STOP
3610 GO TO 3570
4000 REM
4010 REM plot omzet grafiek
4020 CLS : PRINT AT 0,10; INVERSE 1;"OMZET#GRAFIEK"
4030 PRINT AT 2,0; INVERSE 1;"JAAR";AT 1,26; INVERSE 1;'GEM
';AT 2,26; INVERSE 1;"OMZET"
4040 IF qm=1 THEN  GO TO 4500
4050 LET qm=1
4060 PRINT AT 2,15; INVERSE 1; INK 3; FLASH 1;"EVEN#GEDULD!
"
4080 LET max=INT (VAL A$(1)): LET min=max
4090 FOR I=2 TO NY
4100 LET n1=INT (VAL A$(I))
```

```
4110 IF n1>max THEN  LET max=n1
4120 IF n1<min THEN  LET min=n1
4130 BEEP .05,0
4140 NEXT I
4150 LET ran=max-min
4160 LET rat=INT (ran/152+.5)
4165 IF rat=0 THEN  LET rat=.001
4170 LET ox=40
4180 FOR I=1 TO NY
4190 LET n1=INT (VAL A$(I)): LET nx=INT ((n1-min)/rat+.5)+4
0
4200 LET C(I)=nx-ox: LET ox=nx
4210 BEEP .05,24
4220 NEXT I
4230 PRINT AT 2,12;"############"
4500 REM
4510 REM zet omzet grafiek op scherm
4520 FOR I=1 TO NY
4530 LET ln=I+3-1: PRINT AT ln,0;Y$(I);AT ln,25;A$(I)
4540 NEXT I
4550 PLOT 40,155: DRAW 151,0: PLOT 40,155: DRAW 0,-155
4560 PLOT 40+C(1),147
4570 FOR I=2 TO NY
4580 DRAW C(I),-8
4590 NEXT I
4600 INPUT "Copie#op#printer#(j/n)#?#";k$
4610 IF k$="j" OR k$="J" THEN  COPY : GO TO 4600
4620 IF k$="n" OR k$="N" THEN  RETURN
4630 IF k$="E" THEN  STOP
4640 GO TO 4600
7000 REM
7010 REM getallen afronden
7030 LET X$="##########": REM 10 spaties
7040 LET W$="1000000": REM 6 decimale plaatsen
7050 LET T$=STR$ (INT VAL S$)
7060 IF ns>0 THEN  LET T$=X$(1 TO ns-LEN T$)+T$+".": GO TO
7080
7070 LET T$=".0"
7080 LET n3=VAL W$(1 TO nd+1): REM nd is aantal decimale pl
aatsen
7090 LET U$=STR$ (INT ((VAL S$-VAL T$)*n3+.5))
7100 IF VAL U$<VAL W$(1 TO nd) THEN  LET U$=W$(2 TO nd-LEN
U$+1)+U$
7110 IF ns=0 THEN  LET T$="."
7120 LET S$=T$+U$
7130 RETURN
8000 REM
8010 REM hoofd loop
8020 LET hm=0: LET cm=hm: LET gm=hm: REM initialiseer scher
m
8030 GO SUB 1000: REM invoer aantal jaren
8040 GO SUB 2000: REM invoer omzet verloop
8050 GO SUB 3000: REM berekening
8060 GO SUB 4000: REM plot grafiek
8070 GO TO 8040
```

7.3 Bezittingenevaluatie
(Copyright Beam Software)

Beschrijving
Dit is een van de drie zakelijke programma's in dit boek. Dit model kan worden gebruikt om persoonlijke bezittingen te catalogiseren en evalueren. Deze evaluatie is nuttig bij het afsluiten of het behandelen van claims van brand- en diefstalverzekering. Het programma zou ook voor geïnvesteerde goederen bij een kleine onderneming kunnen worden gebruikt.
Elke bezitting wordt geëvalueerd op oorspronkelijke kostprijs, verkoopwaarde en kosten voor vervanging.
De verkoopwaarde wordt berekend door middel van lineaire afschrijving, en de vervangingskosten zijn op basis van het huidige inflatiepercentage. Als basis zou ook een vastgestelde dagwaardeschaal kunnen worden gebruikt.

Gebruik van het programma
Alle invoer en uitvoer loopt weer via het scherm. Voordat de gegevens over persoonlijke bezittingen kunnen worden ingevoerd, moeten er eerst wat algemene gegevens bekend zijn.

Algemene gegevens	
naam	20 karakters
huidig jaar	4 karakters (= >1984)
huidige inflatie	< 100%
aantal items	< 100

Item gegevens	
omschrijving	18 karakters
plaats	10 karakters

aankoopdatum	4 karakters
levensduur	< 100 jaar
aankoopsom	getal

Programma-opbouw

```
Initialisatie
  initialiseer string-variabelen
  invoer algemene gegevens
  initialiseer arrays voor
    omschrijving
    plaats
    aankoopdatum
    levensduur

Invoer item gegevens
  zet item scherm op
  invoer item data-velden
  bereken verkoopwaarde
  bereken vervangingskosten

Hoofdlus
  invoer item gegevens
    totdat alle items ingevuld zijn of toets '*' ingedrukt
  zet beginkop op scherm
  zet eindkop op scherm
  stop
```

Opmerkingen
Alle waarden die in dit programma worden gebruikt zijn afgerond op hele guldens. Dit is gezien het feit dat er met lange termijn bezittingen gewerkt wordt volkomen verantwoord. De gebruiker zou het als een uitdaging kunnen beschouwen om de volgende verbeteringen in het programma aan te brengen.
* alle item-gegevens komen weer in volgorde op het scherm,
* alle items kunnen op PLAATS hergroepeerd worden

Programma

```
 100 REM evaluatie
 110 REM bezittingen evaluatie
 120 GO TO 9000
1000 REM
1010 REM zet beginkop op scherm
1020 REM
1040 CLS : PRINT AT 1,6;"BEZITTINGEN#EVALUATIE": PRINT AT 2
,6;"-- --------------------"
1050 PRINT AT 4,3;'NAAM:#"
```

```
1060 PRINT AT 6,3;"HUIDIG#JAAR##:#"
1070 PRINT AT 8,3;'HUIDIGE#INFLATIE:###.##%"
1080 PRINT AT 10,3;"AANTAL#ITEMS###:#"
1090 PRINT AT 12,1;'TOTAAL": PRINT AT 13,1;"-------"
1100 PRINT AT 14,3;"VERVANGINGSKOSTEN#F1'
1110 PRINT AT 15,3;"AANKOOP##WAARDE'
1120 PRINT AT 16,25;"----------'
1130 PRINT AT 17,3;"VERSCHIL";AT 17,21;"FI"
1140 PRINT AT 18,23;"=========="
1150 PRINT AT 19,3;'pcVERSCHIL";AT 19,14;"##.##%'
1160 PRINT AT 21,3;'TOT#VERKOOPWAARDE#F1"
1170 RETURN
5000 REM
5010 REM item routine
5020 REM item scherm
5030 FOR I=1 TO NI
5040 CLS : PRINT AT 1,8;"ITEM#GEGEVENS": PRINT AT 2,8;"----
---------"
5050 PRINT AT 4,1;"OMSCHRIJVING:"
5060 PRINT AT 6,1;"PLAATS:"
5070 PRINT AT 8,1;"AANKOOPDATUM:";AT 8,21;'DUUR:    jr"
5080 PRINT AT 10,1;"AANKOOPSOM:#F1"
5090 PRINT AT 13,1;"HUIDIGE': PRINT AT 14,1;'--------"
5100 PRINT AT 16,3;"VERKOOPWAARDE";AT 16,21;"F1'
5110 PRINT AT 18,3;"VERVANGINGSKOSTEN#F1'
5120 PRINT AT 4,14; INVERSE 1;"###############'
5130 PRINT AT 6,14; INVERSE 1;"###########'
5140 PRINT AT 8,16; INVERSE 1;'####"
5150 PRINT AT 8,27; INVERSE 1;"##"
5160 PRINT AT 10,16; INVERSE 1;"######'
5170 REM invoer item gegevens
5180 PRINT AT 4,0; FLASH 1;">"
5190 INPUT D$(I): IF D$(I,1)="*" THEN  GO TO 5410
5200 PRINT AT 4,14; INVERSE 1;D$(1)
5210 PRINT AT 4,0;"#": PRINT AT 6,0; FLASH 1;">"
5220 INPUT L$(I): IF L$(I)="*" THEN  GO TO 5410
5230 PRINT AT 6,14; INVERSE 1;L$(I)
5240 PRINT AT 6,0;"#": PRINT AT 8,0; FLASH 1;">"
5250 INPUT T$(I): IF VAL T$(I,1 TO 4)<VAL C$(1 TO 4) THEN
GO TO 5250
5260 PRINT AT 8,16; INVERSE 1;T$(I)
5270 PRINT AT 8,0;'#': PRINT AT 8,20; FLASH 1;">"
5280 INPUT L: IF L<0 THEN  GO TO 5280
5290 LET L=INT L: PRINT AT 8,27; INVERSE 1;L
5300 PRINT AT 8,20;'#': PRINT AT 10,0; FLASH 1;">"
5310 INPUT O: IF O<0 THEN  GO TO 5310
5320 LET O=INT O: PRINT AT 10,16; INVERSE 1;O
5330 PRINT AT 10,0;"#"
5340 REM bereken verkoopwaarde
5350 LET yu=VAL  C$-VAL T$(1): LET yu=(yu<L)*yu+(yu>=L)*L: L
ET rv=INT ((L-yu)*O/L)
5360 LET TC=TC+rv: PRINT AT 16,23+7-LEN (STR$ rv);rv
5370 LET rc=INT (O*(1+LI/100) yu): LET TR=TR+rc: PRINT AT 1
8,21+7-LEN (STR$ rc);rc
5380 LET TO=TO+O
```

```
5390 IF INKEY$="" THEN  GO TO 5390
5400 NEXT I
5410 RETURN
6000 REM
6010 REM zet eindprop op scherm
6020 PRINT AT 4,9; INVERSE 1;N$
6030 PRINT AT 6,17; INVERSE 1;C$
6040 PRINT AT 8,20; INVERSE 1;INT (LI*100)/100
6050 PRINT AT 10,20; INVERSE 1;N1
6060 PRINT AT 14,24+8-LEN (STR$ TR);TR
6070 PRINT AT 15,24+8-LEN (STR$ TO);TO
6080 LET d1=ABS (TR-TO): PRINT AT 17,24+8-LEN (STR$ d1);d1
6090 IF TR<TO THEN  PRINT AT 17,21;"-": PRINT AT 19,10;"-"
6100 PRINT AT 19,14;INT ((d1/TO)*10000)/100;"%"
6110 PRINT AT 21,24+8-LEN (STR$ TC);TC
6120 IF INKEY$<>"*" THEN  GO TO 6120
6130 RETURN
8000 REM
8010 REM initialise data
8020 DIM C$(4): DIM N$(20): LET N$="#": LET CY=0: LET LI=0:
 LET N1=0: LET TR=0: LET TO=0: LET TC=0
8030 GO SUB 1000
8040 PRINT AT 4,9; INVERSE 1;"######################"
8050 PRINT AT 6,17; INVERSE 1;"####"
8060 PRINT AT 8,20; INVERSE 1;"##";AT 8,23;"##"
8070 PRINT AT 10,20; INVERSE 1;"##"
8080 REM invoer velden
8090 PRINT AT 4,2; FLASH 1;">"
8100 INPUT N$: IF N$="#" THEN  GO TO 8100
8110 PRINT AT 4,9; INVERSE 1;N$(1 TO 20)
8120 PRINT AT 4,2;"#": PRINT AT 6,2; FLASH 1;">"
8130 INPUT C$: IF VAL C$<1984 THEN  GO TO 8130
8140 PRINT AT 6,17; INVERSE 1;C$
8150 PRINT AT 6,2;"#"
8160 PRINT AT 8,2; FLASH 1;">"
8180 INPUT LI: IF LI>100 OR LI<0 THEN  GO TO 8180
8190 PRINT AT 8,20;"EXT 0 EXT SHI /"+STR$ (INT (LI*100)/100
)+"EXT 7 EXT SHI 0"
8210 PRINT AT 8,2;"#": PRINT AT 10,2; FLASH 1;">"
8220 INPUT NI: PRINT AT 10,20;"EXT 0 EXT SHI /"+STR$ NI+"EX
T 7 EXT SHI 0"
8230 PRINT AT 10,2;"#"
8240 PRINT AT 12,9; FLASH 1; INK 3;"EVEN#GEDULD!"
8250 DIM D$(NI,18)
8260 DIM L$(NI,10)
8270 DIM T$(NI,4)
8280 DIM L(NI)
8290 DIM O(NI)
8300 FOR I=1 TO NI
8310 LET D$(I)="#": LET L$(I)="#": LET T$(I)="#"
8320 LET L(I)=0: LET O(I)=0
8330 NEXT I
8340 PRINT AT 12,9; INK 3;"OK#LAAK!####": FOR I=1 TO 50: N
EXT I
8350 LET TR=0: LET TO=TR: LET TC=TR
```

```
8360 RETURN
9000 REM
9010 REM hoofd loop
9020 GO SUB 8000: REM invoer begingegevens
9030 GO SUB 5000: REM invoer item
9040 REM eindkop gegevens
9050 GO SUB 1000
9060 GO SUB 6000
9070 PRINT AT 12,9; FLASH 1; INK 1;"PROGRAMMA#IS#KLAAR"
9080 STOP
```

8. Strategische programma's

8.1 Schaken

(Copyright Clifford Ramshaw en Beam Software)

Met dit programma kunt u de computer uitdagen voor een spelletje schaak. Maar wees gewaarschuwd: de computer is een slechte tegenstander en hij is ongelooflijk traag. Desalniettemin, dit programma heeft een mooie grafische vormgeving en een ieder die graag schaakt zou zonder al te veel moeite het peil van de computer kunnen verbeteren. Het programma zou ook nog omgebouwd kunnen worden, zodat twee personen een partij kunnen schaken. De keuze is aan u.

Dit programma is een uitstekend voorbeeld van wat mogelijk is op de Spectrum. Schaken wordt algemeen beschouwd als een moeilijk spel voor computers, en hoewel dit programma geenszins pretendeert goed te kunnen schaken, laat het toch zien wat vrij eenvoudig met Spectrum-Basic bereikt kan worden.

Als het programma wordt opgestart, dan zet de computer de variabelen op, tekent het bord en de stukken, en vraagt u om een zet in te tikken. U speelt met wit (bovenaan) en geeft een zet op met een letter tussen 'a' en 'h', gevolgd door een cijfer tussen '1' en '8'. De computer controleert niet of uw zet legaal is, wees dus voorzichtig (mits u natuurlijk niet opzettelijk vals wilt spelen).

Programma-opbouw

Regel	Beschrijving
1 - 4	zet scherm op definieer enkele variabelen die gebruikt worden om ruimte te besparen in de data-statements die gebruikt worden om de grafische schaakstukken te definieren
1 - 270	gebruik de data-statements in regel 30-220 om stukken als vrij programmeerbare karakters op te zetten
280 - 315	teken leeg bord op scherm
318 - 360	dimensioneer de bord-array en de arrays voor de zetmogelijkheden van de stukken vul de bord array in
370	roep de subroutine aan die de stukken op het scherm zet
380 - 500	vul de zetmogelijkheden-array in
600 - 630	Hoofdlus: 1 haal zet speler op haal zet computer op ga naar 1

800 - 900	invoer spelerzet
1000 - 1010	zet stuk 'c' op bordpositie x, y
3000 - 3150	voer spelerzet uit op scherm
4000 - 4410	voer computerzet uit
5000 - 5090	test of computer schaakmat staat

Programma

```
  1 REM schaal
  2 PAPER 7: BORDER 7: CLS
  3 LET p=248: LET t=31: LET a=128: LET g=13: LET h=160
  4 LET b=192: LET d=224: LET e=240: LET f=15
  5 LET x=0: LET y=1: LET z=3: LET c=7: LET i=160
  6 LET f$="": DIM A$(1,1)
  7 LET kx=5: LET ky=8: LET q=0
 10 FOR m=USR "a" TO USR "t"+7
 20 READ n: POKE m,n: NEXT m
 30 DATA x,x,x,x,x,x,y,z
 40 DATA x,x,x,x,x,x,a,b
 50 DATA c,c,z,y,z,c,x,x
 60 DATA d,d,b,a,b,d,x,x
 70 DATA y,c,c,y,g,f,f,c
 80 DATA a,d,d,e,176,e,e,d
 90 DATA z,z,z,y,c,f,31,x
100 DATA b,b,b,a,d,e,p,x
110 DATA 4,5,2,y,z,c,c,c
120 DATA 32,h,64,a,b,d,d,d
130 DATA z,y,y,y,z,c,f,x
140 DATA b,a,a,a,b,d,e,x
150 DATA y,z,c,c,14,g,c,z
160 DATA a,b,h,112,e,d,d,b
170 DATA x,x,x,x,x,5,5,c
180 DATA x,x,x,x,x,h,h,d
190 DATA x,x,x,y,z,c,c,z
200 DATA x,x,a,b,d,e,p,152
210 DATA y,y,y,z,c,c,f,x
220 DATA a,a,a,b,d,d,e,x
230 DIM t$(7,2): DIM b$(7,2)
240 FOR x=1 TO 7: READ t$(x),b$(x): NEXT x
250 DATA "GRA AB GRU","GRA CD GRU","GRA DK GRU","GRA ST GR
U"
260 DATA "GRA MN GRU","GRA FL GRU","GRA OP GRU","GRA CD GR
U"
270 DATA "GRA IJ GRU","GRA FL GRU","GRA EF GRU","GRA GH GR
U","##","##"
280 LET i=2: LET p=4: PRINT "##A#D#C#D#E#F#G#H"
290 PRINT : FOR x=0 TO 7: FOR y=0 TO 1: PRINT ;"##";
300 FOR z=0 TO 3: PRINT  PAPER p;"##"; PAPER i;"##";
```

```
 310 NEXT z: PRINT : NEXT y: LET t=1: LET 1=p: LET p=t: NEX
T x
 315 FOR y=1 TO 8: PRINT AT y+y,0;y: NEXT y
 320 DIM b(8,8): FOR y=1 TO 8: FOR x=1 TO 8
 330 LET z=(y=8)*-b(x,1)+(y=2)*-1+(y=7)
 340 IF y=1 THEN  READ z
 350 LET b(x,y)=z: NEXT x: NEXT y
 360 DATA -4,-2,-3,-5,-6,-3,-2,-4
 370 GO SUB 2000
 380 DIM d(6,8,2): DIM p(6)
 390 FOR x=1 TO 6: LET p(x)=2: FOR y=1 TO 8
 400 LET d(x,y,2)=((y=3) OR (y=8))+((y=3) AND (y=7))*-1
 410 LET d(x,y,1)=((y=1) AND (y<5))+(y=5)*-1
 420 NEXT y: IF x<>2 THEN  GO TO 460
 430 FOR y=1 TO 4: READ d(x,y,1),d(x,y,2): LET d(x,y+4,1)=-
d(x,y,1): LET d(x,y+4,2)=-d(x,y,2)
 440 NEXT y
 450 DATA -2,1,-1,2,1,2,2,1
 460 IF x=1 THEN  LET p(x)=8
 470 IF x=4 THEN  LET p(x)=1
 480 NEXT x
 490 DIM c$(5): FOR x=1 TO 5: READ c$(x): NEXT x
 500 DATA "s","p","I","t","#"
 600 GO SUB 3000
 610 PRINT AT 8,22; FLASH 1;"denkdenk";
 620 GO SUB 4000
 630 GO TO 600
 800 LET A$(1)=INKEY$: IF A$(1)="p" THEN  GO TO 900
 805 IF (A$(1)="") OR (A$(1)<"a") OR (A$(1)>"h") THEN  GO T
O 800
 810 PRINT A$(1);"#";
 820 LET X1=CODE A$(1)-96
 830 LET A$(1)=INKEY$: IF (A$(1)="") OR (A$(1)<"1") OR (A$(
1)>"8") THEN  GO TO 830
 840 PRINT A$(1);
 850 LET Y1=CODE A$(1)-48
 860 RETURN
 900 LET X1=999: LET Y1=0: RETURN
1000 LET z1=INT ((ATTR (y+y,x+x))/8): IF z1>=8 THEN  LET z1
=z1-8
1010 PRINT  BRIGHT 1; PAPER z1; INK 1;AT y+y,x+x;t$(c);AT y
+y+1,x+x;b$(c);
1020 RETURN
2000 FOR y=1 TO 8: FOR x=1 TO 8
2010 LET z=b(x,y): IF z=0 THEN  GO TO 2040
2020 LET c=7: IF z>0 THEN  LET c=0
2030 LET c=ABS z: GO SUB 1000
2040 NEXT x: NEXT y
2050 RETURN
3000 IF f$="1" THEN  PRINT AT 20,22;"SCHAAKMAT": STOP
3010 PRINT AT 3,22;"SPELER": PRINT AT 4,22;"van##";: GO SUB
 800: IF X1=999 THEN  STOP
3020 LET X=X1: LET Y=Y1
3030 PRINT AT 5,22;"naar#";: GO SUB 800: LET XB=X1: LET YB=
Y1
```

```
3040 IF b(XB,YB)>0 THEN  LET p(b(XB,YB))=p(b(XB,YB))-1
3050 LET b(XB,YB)=b(X,Y)
3060 LET b(X,Y)=0: LET i=7
3070 IF b(XB,YB)<-1 OR YB<8 THEN  GO TO 3120
3080 PRINT AT 6,19;"Nieuw#stuk#";
3090 LET A$(1)=INKEY$
3100 LET a=0: FOR z=1 TO 5: IF c$(z)=A$(1) THEN  LET a=-z
3110 NEXT z: IF a=0 THEN  GO TO 3090
3115 PRINT A$(1);: LET b(XB,YB)=a
3120 LET c=7: GO SUB 1000
3130 LET c=-b(XB,YB): LET x=XB: LET y=YB: GO SUB 1000
3140 PRINT AT 3,22;"##########";AT 4,22;"########";AT 5,22;"
########";AT 6,19;"############";
3150 RETURN
4000 LET xb=0: LET yb=xb: LET db=xb: LET c1=1: LET bp=yb: L
ET x=9
4010 LET ax=kx: LET ay=ky: LET g$="": GO SUB 5010: LET q$=+
$
4020 FOR y=1 TO 8: FOR x=1 TO 8: LET tb=0: LET cz=1: LET ty
=0: LET tx=0: LET d=0
4030 IF b(x,y)<1 THEN  GO TO 4370
4040 LET p=b(x,y): FOR i=1 TO 8: LET dy=d(p,i,2): LET dx=d(
p,i,1): LET ax=kx: LET ay=ky: LET c1=1: LET po=0
4050 IF x+dx<1 OR x+dx>8 OR y+dy<1 OR y+dy>8 THEN  GO TO 43
00
4060 IF p>2 AND p<6 THEN  GO TO 4210
4070 IF dy>-1 AND p=1 THEN  GO TO 4300
4080 IF y>1 OR p>1 THEN  GO TO 4130
4090 LET p(5)=p(5)+1
4100 LET b(x,y)=5
4120 GO TO 4300
4130 IF dx<>0 AND b(x+dx,y+dy)>-1 THEN  GO TO 4300
4140 IF p=6 THEN  LET kx=x: LET ky=y: LET ax=x+dx: LET ay=y
+dy: GO SUB 5000: IF f$="1" THEN  GO TO 4300
4150 IF dx=0 AND b(x,y+dy)<0 THEN  GO TO 4300
4160 LET po=8-p-b(x+dx,y+dy)*3
4170 IF y<>7 OR dx<>0 OR b(x,5)<>0 OR p>1 THEN  GO TO 4300
4180 IF g$="1" THEN  GO SUB 5000: IF f$="" THEN  GO TO 4300

4190 IF g$="1" OR RND>.3 THEN  LET c1=2
4200 GO TO 4300
4210 LET x1=x: LET y1=y
4220 IF p=3 AND INT (i/2)*2<>i THEN  GO TO 4300
4230 IF p=4 AND INT (i/2)*2=i THEN  GO TO 4300
4240 IF x1+dx<1 OR x1+dx>8 OR y1+dy<1 OR y1+dy>8 THEN  LET
c1=c1-1: GO TO 4300
4250 IF b(x1+dx,y1+dy)>0 THEN  LET c1=c1-1: GO TO 4300
4260 LET po=8-p+INT (RND*3)-b(x1+dx,y1+dy)*3
4270 IF g$="1" THEN  GO SUB 5000: IF f$="" THEN  LET po=po+
50: GO TO 4340
4280 IF b(x1+dx,y1+dy)<0 THEN  GO TO 4300
4290 LET c1=c1+1: LET x1=x1+dx: LET y1=y1+dy: GO TO 4240
4300 IF po=0 THEN  GO TO 4340
4310 IF g$="1" THEN  GO SUB 5000: IF f$="" THEN  LET po=po+
50: GO TO 4340
```

```
4320 IF g$="" THEN  GO SUB 5000: IF f$="1" THEN  LET po=0:
GO TO 4340
4330 IF po<=tb THEN  LET ax=x+dx*c1: LET ay=y+dy*c1: GO SUB
 5000: IF f$="1" THEN  LET po=po-p*2
4340 IF po>tb THEN  LET tb=po: LET tx=x: LET ty=y: LET d=i:
 LET cz=c1
4350 NEXT i
4360 IF tb<bp OR (tb=bp AND (RND<.9 OR (ty<yb AND RND <.5)))
 THEN  LET bp=tb: LET xb=tx: LET yb=ty: LET c1=cz: LET db=d

4370 NEXT x: NEXT y: LET p=b(xb,yb): LET x1=xb+d(p,db,1)*c1
: LET y1=yb+d(p,db,2)*c1: IF p=6 THEN  LET ix=x1: LET iy=y1

4380 LET ax=ix: LET ay=iy: LET b(x1,y1)=b(xb,yb): LET b(xb,
yb)=0: LET q=1: GO SUB 5010: LET q=0
4385 PRINT AT 8,22;'#########";AT 9,22;'COMPUTER';AT 10,22;"
van##";CHR$ (xb+95);"-";CHR$ (yb+48);
4386 PRINT AT 11,22;"naar#";CHR$ (x1+95);"-";CHR$ (y1+48);
4390 LET i=0: LET c=7: LET x=xb: LET y=yb: GO SUB 1000
4400 LET c=b(x1,y1): LET x=x1: LET y=y1: GO SUB 1000
4410 RETURN
5000 LET pc=b(x+dx*c1,y+dy*c1): LET b(x+dx*c1,y+dy*c1)=b(x,
y): LET b(x,y)=0
5010 LET f$="": FOR r=1 TO 6: FOR j=1 TO 8: LET d1=d(r,j,1)
: LET d2=d(r,j,2): LET l=1
5020 IF ax+d1*l<1 OR ax+d1*l>8 OR ay+d2*l<1 OR ay+d2*l>8 TH
EN  GO TO 5060
5030 IF r=1 AND d2<1 AND q=0 THEN  GO TO 5060
5040 IF b(ax+d1*l,ay+d2*l)=r THEN  LET f$="1": GO TO 5070
5050 IF r>2 AND r<6 AND b(ax+d1*l,ay+d2*l)=0 THEN  LET l=l+
1: GO TO 5020
5060 NEXT j: NEXT r
5070 IF x>8 THEN  GO TO 5090
5080 LET b(x,y)=b(x+dx*c1,y+dy*c1): LET b(x+dx*c1,y+dy*c1)=
pc
5090 RETURN
```

8.2 Dammen

Dit programma is een combinatie van Basic en machinetaal en speelt een redelijke partij dammen.

De computer tekent een dambord op het scherm met uw schijven (rood) onderaan het scherm en de zijne (zwart) bovenaan. De speler begint altijd en geeft zijn zet op als één 'van waarde' waardoor de computer weet met welke schijf u wilt bewegen, en één 'naar waarde', die aangeeft naar welk vak op het bord de schijf moet gaan.

De vakken worden aangeduid door een letter en een cijfer. De letters lopen van 'a' tot 'h' en geven de kolommen aan, de cijfers staan voor de rijen en lopen van '1' tot '8'. Een openingszet zou bijvoorbeeld 'VAN a6, NAAR b5' kunnen zijn (u hoeft alleen maar 'a6' en 'b5' in te tikken).

U kunt te allen tijde het spel stoppen door een '0' in te tikken als er een zet opgevraagd wordt.
Van alle zetten wordt gecontroleerd of ze legaal zijn, zodat u niet vals kunt spelen. Slaan is niet verplicht, hoewel de computer hieraan de voorkeur geeft, en meervoudige sprongen zijn niet toegestaan.
Als u een schijf op de laatste rij krijgt dan heeft u een dam. Deze mogen maar over een enkel vak bewegen en maar een schijf tegelijk slaan, maar ze kunnen wel voor- en achterwaarts bewegen.

Hoe het programma werkt

Het BASIC-gedeelte van het programma voert alle huishoudtaken uit. Het zet het bord op, regelt de invoer en uitvoer van uw zetten, en houdt het scherm bij. Een kort machinetaalprogramma wordt aangeroepen om de zet van de computer te berekenen. Dit is nodig om niet eeuwig op zijn zet te hoeven wachten. Een uitdraai van het machinetaalgedeelte is afgedrukt voor hen die hierin geïnteresseerd zijn.
Maakt u zich geen zorgen als u geen verstand van machinetaal heeft, want dit gedeelte wordt verzorgd door de data-statements in regel 9000 - 9450. Tik ze wel nauwkeurig in!
In het machinetaalgedeelte houdt de computer een kopie van het dambord bij in de vorm van een serie bytes. Daarom worden veranderingen in de opstelling aangebracht door middel van een PEEK en POKE opdracht in plaats van gebruik te maken van een array. Het is namelijk veel eenvoudiger voor machinetaal om geheugenadressen te hanteren dan arrays.

Het programma:

Regel	Beschrijving
10 - 20	brengen topgeheugen (RAMTOP) omlaag ter bescherming machinetaalgedeelte dat met subroutine 9000 hier neergezet wordt
30 - 79	zet de scherm attributen bouw kopie bord in geheugen
80 - 160	definieer de vrij programmeerbare karakters nodig voor de schijven zet leeg scherm op bord
170 - 230	zet schijven op bord in beginopstelling
300 - 320	Hoofdlus 1 haal zet speler op haal zet computer op ga naar 1

1000 - 1060 zet schijf op bord op positie x,y in kleur c
schijf is gedefinieerd door p
0 = leeg
1 = normale schijf
2 = dam

2000 - 2540 invoer zet speler

uitsplitsing:
2000-2020 melding op scherm
2030-2050 haal 'van'waarde op, legaal?
2060-2110 haal 'naar'waarde op, legaal?
2120-2200 als slag, doe zet speler
2500-2540 als normaal, doe zet speler

3000 - 3110 roept machinetaal aan voor zet computer en voert die zet uit op het bord

3500 zet x,y-waarden computerzet op scherm

3900 - 3904 programma springt hier naar toe als computer heeft verloren

4000 - 5010 invoer x,y-waarden zet speler
als eerste karakters = 0' dan wordt het spel gestopt en de speler gevraagd of hij nog een keer wil spelen

9000 - 9020 machinetaalgedeelte wordt uit data-statements gelezen en boven RAMTOP gezet

9100 - 9450 hier staan de data-statements die de machinetaal bevatten

Programma

```
 1 REM dam
10 CLEAR 32019
20 GO SUB 9000
30 RESTORE 70
40 BORDER 7: PAPER 7: INK 0: CLS
50 FOR x=32420 TO 32420+89
60 READ a: POKE x,a: NEXT x
70 DATA 255,255,255,255,255,255,255,255,255
71 DATA 0,1,0,1,0,1,0,1,255
72 DATA 1,0,1,0,1,0,1,0,255
73 DATA 0,1,0,1,0,1,0,1,255
74 DATA 0,0,0,0,0,0,0,0,255
75 DATA 0,0,0,0,0,0,0,0,255
```

```
  76 DATA 2,0,2,0,2,0,2,0,255
  77 DATA 0,2,0,2,0,2,0,2,255
  78 DATA 2,0,2,0,2,0,2,0,255
  79 DATA 255,255,255,255,255,255,255,255,255
  80 FOR x=USR 'a' TO 63+USR "a"
  90 READ a: POKE x,a: NEXT x
 100 DATA 0,7,31,63,63,127,127,127
 101 DATA 0,224,248,252,252,254,254,254
 102 DATA 127,127,127,63,63,31,7,0
 103 DATA 254,254,254,252,252,248,224,0
 104 DATA 0,7,31,60,56,116,98,97
 105 DATA 0,224,248,60,28,46,70,134
 106 DATA 97,98,116,56,60,31,7,0
 107 DATA 134,70,46,28,60,248,224,0
 110 PRINT "##A#B#C#D#E#F#G#H": PRINT
 115 LET d=3: LET l=5
 120 FOR y=1 TO 8: PRINT AT y+y,0;y;"#";
 130 FOR z=1 TO 2: FOR x=1 TO 4
 140 PRINT  PAPER d;"##"; PAPER l;'##";: NEXT x: PRINT : PR
INT "##';: NEXT z
 150 LET t=d: LET d=l: LET l=t
 160 NEXT y
 170 LET p=1
 180 FOR x=2 TO 8 STEP 2
 190 LET c=0: LET y=1: GO SUB 1000: LET y=3: GO SUB 1000
 200 LET c=2: LET y=7: GO SUB 1000: NEXT x
 210 FOR x=1 TO 7 STEP 2
 220 LET c=0: LET y=2: GO SUB 1000
 230 LET c=2: LET y=6: GO SUB 1000: LET y=8: GO SUB 1000: N
EXT x
 300 GO SUB 2000
 310 GO SUB 3000
 320 GO TO 300
1000 PAPER 1: INK c: PRINT AT y+y,x+x;
1005 IF p=0 THEN  PRINT "##";
1010 IF p=1 THEN  PRINT "GRA AB GRU";
1020 IF p=2 THEN  PRINT "GRA EF GRU";
1030 PRINT AT y+y+1,x+x;
1035 IF p=0 THEN  PRINT '##";
1040 IF p=1 THEN  PRINT "GRA CD GRU";
1050 IF p=2 THEN  PRINT "GRA GH GRU":
1060 RETURN
2000 PAPER 7: INK 0: PRINT AT 6,22;"SPELER"
2010 BEEP .5,10: PRINT AT 7,22;"van##:###";
2020 PRINT AT 8,22;"naar#:###";
2030 LET l=7: GO SUB 4000: LET fx=x: LET fy=y
2040 LET fp=32419+9*fy+fx: LET f=PEEK fp
2050 IF f<>2 AND f<>130 THEN  GO TO 2010
2060 LET l=8: GO SUB 4000: LET tx=x: LET ty=y
2070 LET tp=32419+9*ty+tx: LET t=PEEK tp
2080 LET dx=tx-fx: LET dy=ty-fy
2090 IF (ABS dx<>1 AND ABS dx<>2) OR ABS dx<>ABS dy THEN  G
O TO 2010
2100 IF (dy=1 OR dy=2) AND f=2 THEN  GO TO 2010
2110 IF ABS dx=1 THEN  GO TO 2500
```

```
2120 IF t<>0 THEN  GO TO 2010
2130 LET jx=fx+dx/2: LET jy=fy+dy/2
2140 LET jp=32419+9*jy+jx: LET j=PEEK jp
2150 IF j<>1 AND j<>129 THEN  GO TO 2010
2160 LET c=2: LET p=0: LET x=fx: LET y=fy: GO SUB 1000: POK
E fp,0
2170 LET p=1: IF f=130 OR ty=1 THEN  LET p=2
2180 LET x=tx: LET y=ty: GO SUB 1000: POKE tp,f: IF ty=1 TH
EN  POKE tp,130
2190 LET p=0: LET x=jx: LET y=jy: GO SUB 1000: POKE jp,0
2200 RETURN
2500 IF t<>0 THEN  GO TO 2010
2510 LET p=0: LET c=2: LET x=fx: LET y=fy: GO SUB 1000: POK
E fp,0
2520 LET p=1: IF f=130 OR ty=1 THEN  LET p=2
2530 LET x=tx: LET y=ty: GO SUB 1000: POKE tp,f: IF y=1 THE
N  POKE tp,130
2540 RETURN
3000 LET mp=USR 32020
3002 PAPER 7: INK 0: PRINT AT 10,22;"COMPUTER"
3003 PRINT AT 11,22;"van##:###";
3004 PRINT AT 12,22;'naar#:###";
3005 FOR i=1 TO 50: NEXT i
3010 IF mp=0 THEN  GO TO 3900
3020 LET fy=INT (mp/256): LET ty=mp-256*fy
3030 LET fx=INT (fy/16): LET fy=fy-16*fx
3040 LET tx=INT (ty/16): LET ty=ty-16*tx
3045 LET x=fx: LET y=fy: LET z=11: GO SUB 3500: LET x=tx: L
ET y=ty: LET z=12: GO SUB 3500
3050 LET p=0: LET x=fx: LET y=fy: GO SUB 1000
3060 LET tp=32419+9*ty+tx: LET t=PEEK tp
3070 LET p=1: IF t=129 THEN  LET p=2
3080 LET c=0: LET x=tx: LET y=ty: GO SUB 1000
3090 IF ABS (tx-fx)<>2 THEN  RETURN
3100 LET x=fx+(tx-fx)/2: LET y=fy+(ty-fy)/2
3110 LET p=0: GO SUB 1000: RETURN
3500 PRINT AT z,28;CHR$ (y+64);CHR$ (y+48): RETURN
3900 PRINT  INK 0;AT 20,0;"i#qeef#op#--#J1J#HEBT#GEWONNEN"

3901 PRINT AT 21,0;'Nog#een#keer#(J/N)'
3902 IF INKEY$="" THEN  GO TO 3902
3903 IF INKEY$="j" THEN  GO TO 30
3904 STOP
4000 PRINT AT 1,28; FLASH 1;'#";CHR$ 8;
4010 LET a$=INKEY$
4015 IF a$="0" THEN  GO TO 5000
4020 IF a$<"a" OR a$>"h" THEN  GO TO 4010
4030 PRINT a$: LET x=CODE a$-96
4040 PRINT AT 1,29; FLASH 1;"#";CHR$ 8;
4050 LET a$=INKEY$
4060 IF a$<"1" OR a$>"8" THEN  GO TO 4050
4070 PRINT a$: LET y=CODE a$-48
4080 RETURN
5000 PRINT  INK 0;AT 20,0;'jammer--#IK#HEB#GEWONNEN'
5010 GO TO 3901
```

```
9000 RESTORE 9100: FOR x=32020 TO 32301
9010 READ a: POKE x,a: NEXT x
9020 RETURN
9100 DATA 175,33,177,125,6,10,119,35
9110 DATA 16,252,6,35,33,174,126,176
9120 DATA 254,1,40,16,254,129,32,24
9130 DATA 17,246,255,205,187,125,17,248
9140 DATA 255,205,187,125,17,8,0,205
9150 DATA 187,125,17,10,0,205,187,125
9160 DATA 35,35,16,219,58,182,125,167
9170 DATA 40,18,42,183,125,237,91,185
9180 DATA 125,229,126,54,0,25,54,0
9190 DATA 25,119,24,19,58,177,125,167
9200 DATA 40,29,42,178,125,237,91,180
9210 DATA 125,126,54,0,229,25,119,235
9220 DATA 225,205,135,125,121,230,15,254
9230 DATA 8,192,26,246,128,18,201,1
9240 DATA 0,0,201,235,205,146,125,79
9250 DATA 235,205,146,125,71,201,229,213
9260 DATA 197,17,164,126,175,237,82,61
9270 DATA 17,9,0,60,237,82,48,251
9280 DATA 71,25,125,60,7,7,7,7
9290 DATA 128,193,209,225,201,0,0,0
9300 DATA 0,0,0,0,0,0,0,205
9310 DATA 19,126,254,255,200,230,127,254
9320 DATA 1,200,254,2,40,78,229,25
9330 DATA 213,17,246,255,205,13,126,40
9340 DATA 39,17,248,255,205,13,126,40
9350 DATA 21,17,8,0,205,5,126,40
9360 DATA 23,17,10,0,205,5,126,40
9370 DATA 15,209,225,34,178,125,237,83
9380 DATA 180,125,62,1,50,177,125,201
9390 DATA 58,177,125,167,40,235,209,225
9400 DATA 201,205,19,126,230,127,254,2
9410 DATA 201,205,19,126,254,130,201,229
9420 DATA 25,126,225,201,229,25,205,19
9430 DATA 126,225,254,0,192,62,1,50
9440 DATA 182,125,34,183,125,237,85,185
9450 DATA 125,201
```

```
                    110 ;
7D14                120         ORG   32020
                    130 ;
0000                140 LEEG    EQU   0          ;CODE LEEG VAK
FFFF                150 NUL     EQU   -1         ;CODE RAND BORD
0001                160 ZWART   EQU   1          ;CODE ZWARTE STEEN
0002                170 WIT     EQU   2          ;CODE WITTE STEEN
0080                180 DAM     EQU   128        ;TEL OP BIJ DAM
                    190 ;
7EA4                200 BORD    EQU   32420      ;BORDADRES
                    210 ;
7D14                220 START   EQU   $
7D14 AF             230         XOR   A
7D15 21B17D         240         LD    HL,ZET     ;WIS OUDE ZETTEN
7D18 0650           250         LD    B,10       ;IN GEHEUGEN
7D1A 77             260 CLEAR   LD    (HL),A
7D1B 23             270         INC   HL
7D1C 10FC           280         DJNZ  CLEAR
                    290 ;
7D1E 0623           300         LD    B,35 ;TEL IE CHECKEN POS.
7D20 21AE7E         310         LD    HL,BORD+10 ;HL=BORD POSITIE
                    320 ;
7D23 7E             330 VOLG    LD    A,(HL)       ;ZOEK STEEN
7D24 FE01           340         CP    ZWART    ;COMPUTER STEEN?
7D26 2810           350         JR    Z,ZOEKST ;JA :=NORMALE STEEN
7D28 FE81           360         CP    ZWART+DAM ;COMPUTER DAM?
7D2A 2018           370         JR    NZ,STOPCH ;NEE>=VOLGENDE ST
7D2C 11F6FF         380         LD    DE,-10   ;TEST ZET ACHTERUIT
7D2F CDB87D         390         CALL  TEST
7D32 11F8FF         400         LD    DE,-8    ;VOOR DAM
7D35 CDB87D         410         CALL  TEST
7D38 110800         420 ZOEKST  LD    DE,8 ;TEST SPRONG VOORUIT
7D3B CDB87D         430         CALL  TEST ;VOOR
7D3E 110A00         440         LD    DE,10    ;OF DAM OF STEEN
7D41 CDB87D         450         CALL  TEST
7D44 23             460 STOPCH  INC   HL    ;VOLGENDE STN (TEST OM
7D45 23             470         INC   HL      ;DE TWEE POSITIES)
7D46 10DB           480         DJNZ  VOLG
                    490 ;
7D48 3AB67D         500         LD    A,(SLAG) ;CHECK OF SLAG
7D4B A7             510         AND   A
7D4C 2812           520         JR    Z,NORZET ;NEE>=TEST GEW.ZET
7D4E 2AB77D         530         LD    HL,(SLAGJ)   ;JA DUS DOE ZET
7D51 ED5BB97D       540         LD    DE,(SLAGP) ;BEPAAL RICHTING
7D55 E5             550         PUSH  HL
7D56 7E             560         LD    A,(HL)
7D57 3600           570         LD    (HL),LEEG    ;LEEG VAK
7D59 19             580         ADD   HL,DE
7D5A 3600           590         LD    (HL),LEEG  ;VERWIJDER STEEN
7D5C 19             600         ADD   HL,DE
7D5D 77             610         LD    (HL),A ;ZET ST IN NIEUW VAK
7D5E 1813           620         JR    ZETGEV ;GEEF ZET AAN BASIC
```

```
7D60                    630 NORZET EQU  $
7D60 3AB17D             640        LD   A,(ZET)   ;CHECK OF ZET
7D63 A7                 650        AND  A
7D64 281D               660        JR   Z,ZETNEE ;NEE>=SPELER WINT
7D66 2AB27D             670        LD   HL,(VAN)    ;DOE ZET OP BORD
7D69 ED5BB47D           680        LD   DE,(RICHT)
7D6D 7E                 690        LD   A,(HL)    ;BEWAAR STEEN
7D6E 3600               700        LD   (HL),LEEG    ;LEEG VAK
7D70 E5                 710        PUSH HL
7D71 19                 720        ADD  HL,DE
7D72 77                 730        LD   (HL),A ;ZET ST IN NIEUW VAK
7D73 EB                 740 ZETGEV EX   DE,HL    ;DE=NIEUWE BORDPOS
7D74 E1                 750        POP  HL    ;HAAL OUDE BORDPOS OP
7D75 CD877D             760        CALL VERT      ;VERTAAL UIT BASIC
7D78 79                 770        LD   A,C
7D79 E60F               780        AND  15      ;CHECK OF DAM HALEN
7D7B FE08               790        CP   8
7D7D C0                 800        RET  NZ    ;NEE>=TERUG NAAR BASIC
7D7E 1A                 810        LD   A,(DE)    ;ZOEK STEEN
7D7F F680               820        OR   DAM       ;MAAK HET DAM
7D81 12                 830        LD   (DE),A
7D82 C9                 840        RET
                        850 ;
7D83 010000             860 ZETNEE LD   BC,0 ;GEEF GEEN ZET MOGE-
7D86 C9                 870        RET  ;LIJK DOOR AAN BASIC
                        880 ;
                        890 ;DEZE SUBROUTINE NEEMT HL EN DE
                        895 ;ALS BORDPOSITIES
                        900 ;EN VERTAALD ZE IN X,Y COORDINATEN
                        905 ;VOOR HET BASICPROGRAMMA
                        910 ;
                        920 ;       HL =B (DIT IS DE X,Y WAARDE VAN
                        930 ;       DE 'VAN' POSITIE COMPUTER ZET)
                        940 ;
                        950 ;       DE>=C (X,Y WAARDE VAN DE 'NAAR'
                        955 ;       POSITIE)
                        960 ;DE X WAARDE STAAT IN DE BOVENSTE HELFT
                        965 ;VAN REGISTER B OF C, DE Y WAARDE STAAT
                        970 ;IN DE ONDERSTE HELFT
                        980 ;
7D87                    990 VERT   EQU  $
7D87 EB                1000        EX   DE,HL
7D88 CD927D            1010        CALL VERTIT    ;VERTAAL DE IN C
7D8B 4F                1020        LD   C,A
7D8C EB                1030        EX   DE,HL
7D8D CD927D            1040        CALL VERTIT    ;VERTAAL HL IN B
7D90 47                1050        LD   B,A
7D91 C9                1060        RET
                       1070 ;
                       1080 ;VERTAAL HL IN X,Y COORDINATEN EN ZET
                       1085 ;RESULTAAT IN REGISTER A
                       1090 ;
                       1100 ;DE X WAARDE (KOLOMNO STEEN OP BORD)
                       1110 ;STAAT IN BOVENSTE HELFT REGISTER A
                       1120 ;DE Y WAARDE (REGELNO STEEN OP BORD)
```

```
                1130 ;STAAT IN ONDERSTE HELFT A
                1140 ;
7D92 E5         1150 VERTIT PUSH HL
7D93 D5         1160        PUSH DE         ;BEWAAR REGISTERS
7D94 C5         1170        PUSH BC
7D95 11A47E     1180        LD   DE,BORD    ;HLPSTEEN ADRES
7D98 AF         1190        XOR  A          ;BORDADRES AFTREKKEN
7D99 ED52       1200        SBC  HL,DE      ;OM VERSCHIL OF BORD
7D9B 7D         1210        DEC  A          ;TE KRIJGEN
7D9C 110900     1220        LD   DE,9       ;LENGTE REGEL
7D9F            1230 CNVL   EQU  $
7D9F 3C         1240        INC  A          ;VERHOOG REGELNO
7DA0 ED52       1250        SBC  HL,DE      ;TREK VERSCHIL AF
7DA2 30FB       1260        JR   NC,CNVL    ;MEER RIJGELS?
7DA4 47         1270        LD   B,A        ;B=REGELNO (Y)
7DA5 19         1280        ADD  HL,DE      ;HAAL KOLOMNO OP
7DA6 7D         1290        LD   A,L        ;A=KOLOMNO (X)
7DA7 3C         1300        INC  A          ;EERSTE KOLOM=1
7DA8 07         1310        RLCA
7DA9 07         1320        RLCA ;IN BOVENSTE HELFT
7DAA 07         1330        RLCA ;REGISTER A
7DAB 07         1340        RLCA
7DAC 80         1350        ADD  A,B        ;TEL OP IN Y
7DAD C1         1360        POP  BC
7DAE D1         1370        POP  DE         ;HAAL REGISTERS OP
7DAF E1         1380        POP  HL
7DB0 C9         1390        RET
                1400 ;
                1410 ;DATA GEBIED VOOR NORMALE ZETTEN
                1420 ;
7DB1 00         1430 ZET    DEFB 0 ;VLAG OM AANTEGEVEN DAT
                1435 ;                 ER EEN ZET GEVONDEN IS
7DB2 0000       1440 VAN    DEFW 0 ;BORDADRES VAN TE ZETTEN
                1445 ;                 STEEN
7DB4 0000       1450 RICHT  DEFW 0 ;RICHTING ZET
                1460 ;
                1470 ;DATA GEBIED VOOR SLAGEN
                1480 ;
7DB6 00         1490 SLAG   DEFB 0 ;VLAG
7DB7 0000       1500 SLAGJ  DEFW 0 ;BORDADRES SLAGSTEEN
7DB9 0000       1510 SLAGR  DEFW 0 ;RICHTING ZET
                1520 ;
                1530 ;
                1540 ;
                1550 ;TEST  SOORT ZET MOGELIJK VOOR STEEN OP
                1555 ;      POSITIE (HL)
                1560 ;      TE ZETTEN NAAR PUS (HL)+(DE)
                1570 ;
                1580 ;      ALS  LEGALE ZET MOGELIJK IS
                1585 ;      DAN SLA ZET OP IN
                1590 ;      DATA GEBIED VOOR NORMALE ZETTEN
                1600 ;
                1610 ;      ALS SLAG MOGELIJK IS, SLA ZET OP
                1620 ;      IN DATAGEBIED VOOR SLAGEN
                1630 ;
```

```
                1640 ;       LEGALE NORMALE ZETTEN DIE
                1645 ;       MOGELIJK VALSTRIK ZIJN (=METEEN
                1650 ;       GESLAGEN KUNNEN WORDEN)
                1660 ;       WORDEN ALLEEN OPGESLAGEN ALS
                1670 ;       GEEN ANDERE ZET MOGELIJK IS
                1680 ;
7DB8            1690 TEST    EQU  $
7DB8 CD137E     1700         CALL BRDSTN
7DBB FEFF       1710         CP   NUL        ;CHECK RAND BORD
7DBD C8         1720         RET  Z          ;JA? =GEEN ZET
7DBE E67F       1730         AND  127
7DC0 FE01       1740         CP   ZWART      ;EIGEN ST? =GEEN ZET
7DC2 C8         1750         RET  Z
7DC3 FE02       1760         CP   WIT        ;CHECK SLAG
7DC5 2846       1770         JR   Z,EVSLAG   ;JA =TEST LEGAAL
                1780 ;
                1790 ;EVALUEER NORMALE ZET
                1800 ;
7DC7 E5         1810         PUSH HL
7DC8 19         1820         ADD  HL,DE
7DC9 D5         1830         PUSH DE
7DCA 11F6FF     1840         LD   DE,-10     ;CHECK VALSTRIKKEN
7DCD CDDD7E     1850         CALL ZOEKWD     ;DAM ACHTER STUK?
7DD0 3827       1860         JR   Z,VAL      ;JA, VALSTRIK
7DD2 11F8FF     1870         LD   DE,-8
7DD5 CDDD7E     1880         CALL ZOEKWD
7DD8 281F       1890         JR   Z,VAL
7DDA 110800     1900         LD   DE,8       ;STAAT ER STEEN
7DDD CD057E     1910         CALL ZOEKW      ;VOOR COMP STEEN?
7DE0 2817       1920         JR   Z,VAL      ;JA, VALSTRIK
7DE2 110A00     1930         LD   DE,10
7DE5 CD057E     1940         CALL ZOEKW
7DE8 280F       1950         JR   Z,VAL
                1960 ;
7DEA D1         1970 SLAOF   POP  DE         ;GEEN VAL DUS SLA
7DEB C1         1980         POP  HL         ;SLA ZET OP
7DEC 227B7D     1990         LD   (VAN),HL
7DEF ED537B7D   2000         LD   (RICHT),DE
7DF3 3E01       2010         LD   A,1
7DF5 32817D     2020         LD   (ZET),A    ;ZET VLAG OP 1
7DF8 C9         2030         RET
                2040 ;
7DF9            2050 VAL     EQU  $
7DF9 3A817D     2060         LD   A,(ZET)    ;VAL, ZOEK ANDERE
7DFC A7         2070         AND  A          ;ZET
7DFD 28EB       2080         JR   Z,SLAOF    ;NEE,= SLA DEZE OP
7DFF D1         2090         POP  DE         ;VERGEET HET ANDERE
7E00 E1         2100         POP  HL
7E01 C9         2110         RET
                2120 ;
7E02            2130 ZOEKW   EQU  $
7E02 CD137E     2140         CALL BRDSTN     ;CHECK OF STEEN OP
7E05 E67F       2150         AND  127        ;(HL)+(DE) IS WIT
7E07 FE02       2160         CP   WIT
```

```
7E0C C9          2170          RET
                 2180 ;
7E0D             2190 ZOEKWD   EQU   $
7E0D CD137E      2200          CALL  BRDSTN    ;CHECK OF STEEN OP
7E10 FE82        2210          CP    WIT+DAM   ;(HL)+(DE) IS
7E12 C9          2220          RET   ;WITTE DAM
                 2230 ;
7E13 E5          2240 BRDSTN   PUSH  HL        ;RETURN STEEN OP
7E14 19          2250          ADD   HL,DE     ;(HL)+(DE)
7E15 7E          2260          LD    A,(HL)
7E16 E1          2270          POP   HL
7E17 C9          2280          RET
                 2290 ;
7E18             2300 CVSLAG   EQU   $         ;CHECK OF LEGALE
7E18 E5          2310          PUSH  HL        ;SLAG
7E19 19          2320          ADD   HL,DE
7E1A CD137E      2330          CALL  BRDSTN    ;STEEN ANDERE KANT
7E1D E1          2340          POP   HL        ;STEEN TE SLAAN
7E1E FE00        2350          CP    LEEG      ;LEEG?
7E20 C0          2360          RET   NZ        ;NEE,=GEEN SLAG
7E21 3E01        2370          LD    A,1       ;JA,=SLA SLAG OP
7E23 327B7D      2380          LD    (SLAG),A
7E26 227B7D      2390          LD    (SLAGJ),HL
7E29 ED537D7D    2400          LD    (SLAGR),DE
7E2D C9          2410          RET
                 2420 ;
```

8.3 Avontuur

Dit is een programma dat u in staat stelt vreemde nieuwe werelden te bezoeken en te ontdekken. U vindt hier schatten en vecht tegen gemene monsters. wilt u ooit levend ontsnappen dan moet u raadsels en problemen oplossen.

Met dit programma kunt u die avonturen op uw Spectrum beleven. De computer beschrijft de locaties die u bezoekt en de dingen die u er ziet. Hij vraagt u ook wat voor actie u wilt ondernemen. De opdrachten die u kunt geven bestaan uit een of twee woorden. U moet altijd een werkwoord gebruiken. om de actie te omschrijven, en vaak heeft u een zelfstandig naamwoord nodig. om het voorwerp te beschrijven waarop de actie betrekking heeft. Enige voorbeelden:

'noord - ga naar het noorden en kijk rond
'pak bijl - als er een bijl aanwezig is daar waar u zich bevindt, dan zal de computer proberen deze voor u te pakken.

In enkele gevallen wordt u om meer informatie gevraagd nadat u een opdracht gegeven heeft. Als u bijvoorbeeld 'vermoord draak' ingetikt heeft, dan zal de computer vragen waarmee u dat wilt doen. In dit geval geeft u de naam van het wapen op, of niets (alleen ENTER intikken) als u de draak met blote handen wilt vermoorden.

Dit programma geeft u niet alleen een avontuur om te beleven, maar maakt het u ook mogelijk om uw eigen avontuurlijke werelden te scheppen.

Hoe wordt het spel gespeeld?
De computer zet een beschrijving van de locatie waar u zich bevindt en een beschrijving van de eventueel aanwezige voorwerpen op het scherm. Daarna wordt er aan u gevraagd wat u wilt doen. U moet met een opdracht zoals boven omschreven antwoorden. De computer zal uw opdracht proberen uit te voeren en als alles gelukt is dan vraagt hij om een nieuwe opdracht. Als uw opdracht echter niet mogelijk is - misschien door een spelfout - of als uw opdracht nu nog niet uitgevoerd mag worden, dan zal de computer dit melden, teruggaan en opnieuw proberen

Enkele opdrachten die u kunt gebruiken.

* noord, zuid, oost, west (om)hoog, (om)laag
  Hiermee kunt u in elke richting bewegen.
* inventariseer - een lijst van alle voorwerpen die u bij zich heeft
* pak - pak een voorwerp op
* val - laat een voorwerp vallen

enz

Als de beschrijving van een voorwerp een bijvoeglijk naamwoord bevat - bijv. 'gemene trol', dan kunt u niet zeggen 'vermoord trol'. U moet dan 'vermoord gemene trol' of alleen maar 'vermoord gemene' gebruiken.
Merk op dat woorden afgekort mogen worden tot 1 of 2 letters, mits hierdoor geen verwarring kan ontstaan. Als een afkorting voor meer dan een woord opgaat dan kiest de computer het eerste woord dat hij tegenkomt.

Als u een avontuur aan het spelen bent dan is het vaak een goed idee om een kaart te tekenen van de wereld die u aan het ontdekken bent. Daar veel van dit soort spelen een doolhof bevatten is het bijna onmogelijk om de weg niet kwijt te raken als u geen goede kaart bij zich heeft.

Hoe het programma werkt

Het programma bestaat uit twee afzonderlijke delen. Eerst krijgt u het databestand wat feitelijk het eigenlijke spel is. Het bevat alle beschrijvingen, voorwerpen en acties waaruit het spel bestaat. Daarna krijgt u het programma, waarmee de speler de mogelijkheid krijgt, om het databestand te communiceren.
Dit bestaat uit het opzetten van het spel, het opvragen van de opdrachten en het uitvoeren van de gevraagde acties.

Het databestand

De fundamentele onderdelen van dit spel zijn de locaties waaruit de wereld is opgebouwd, een kaart die de onderlinge relaties tussen de locaties beschrijft, de voorwerpen die in deze wereld voorkomen en de acties die de speler mag uitvoeren. We zullen nu uitvoerig ingaan op hoe deze onderdelen zijn opgebouwd.

Locaties

Voor iedere locatie is een beschrijving nodig om de speler te geven, zodat hij weet waar hij is, en een lijst van locaties waarheen hij vanaf hier kan gaan. Verder is het vaak nodig om een bepaalde voorwaarde te stellen waaraan voldaan moet worden om de locatie te kunnen betreden. Dit kan van alles zijn, van het er zeker van zijn dat de speler een lamp bij zich heeft als hij binnenkomt, tot het laten uitvoeren van een bepaalde actie bij binnenkomst (bijv. een monster laten aanvallen als de speler een bepaald voorwerp niet bij zich heeft).
In dit programma heeft elke locatie een eigen nummer dat de computer gebruikt om die locatie aan te roepen. De nummers beginnen met 1 en lopen in stappen van 1 op tot de laatste locatie, in dit geval nummer 30.

De locaties staan in het programma opgeslagen in data-statements in het volgende formaat:

- regelnummer DATA string, N,Z,O,W,H,L, voorwaarde
- het regelnummer is het locatienummer + bepaalde constante (9000 in dit spel)
- de string is de beschrijving van de locatie
- de waarden N,Z,O,W,H,L zijn de nummers van de locaties waar de speler uitkomt als hij de desbetreffende richting uitgaat (noord, zuid, (om)hoog, (om)laag). Als hier een 0 staat loopt de desbetreffende richting dood.
  de voorwaarde is het nummer van de voorwaarde, waaraan voldaan moet worden om DEZE locatie binnen te mogen. De voorwaarden bestaan uit stukjes programma die staan op het voorwaardenummer + constante (7100 in dit spel).

Voorwerpen

Wat u moet weten over elk voorwerp is de beschrijving, waar het voorwerp zich bevindt, het soort voorwerp (is het bijv. een monster of een schat), hoe sterk het is, en om een grotere flexibiliteit te krijgen, een voorwaarde waaraan voldaan moet worden wil de speler dit voorwerp kunnen pakken (bijv. de voorwaarde dat de speler vermoord wordt als hij een verboden voorwerp probeert te pakken).

De computer herkent voorwerpen door hun nummers, net als locaties. De manier waarop de data voor voorwerpen is opgeslagen is iets anders dan bij locaties. Elk voorwerp heeft twee regels in het programma voor de beschrijving.

De eerste is

regelnummer DATA-string

waarin

regelnummer	= voorwerpnummer + constante (hier 8000)
string	– voorwerp beschrijving

De tweede is

regelnummer DATA-soort, kracht, voorwaarde

waarin

regelnummer	= voorwerpnummer + constante (hier 8500)
soort	– 0 voor normaal voorwerp > 0 voor een schat, dit getal is tevens de score voor het vinden van de schat < 0 voor een monster
kracht	deze wordt op twee manieren berekend - bij een wapen is het het getal dat bij de kracht van de speler wordt opgeteld als hij dit wapen gebruikt - bij een monster is het de grootte van de kracht die bij een gevecht tegen de speler wordt ingezet
voorwaarde	= voorwaarde waaraan voldaan moet worden wil de speler dit voorwerp op kunnen pakken

Ten slotte hebben we nog de locatie nodig voor elk voorwerp; deze wordt opgeslagen in een 'array'. De beginlocaties voor ieder voorwerp en het aantal voorwerpen staan opgeslagen in een DATA-statement regel 8000, en op basis hiervan wordt de locatie-array opgebouwd.

De waarden die voor deze locaties worden gebruikt:

-1 - voor voorwerpen die u niet kunt zien of gebruiken, bijv. dode dieren of schatten die terug gezet zijn in de beginlocatie.

0 - voorwerpen die de speler bij zich heeft

> 0 - alle andere voorwerpen en dit getal is tevens het locatienummer van het voorwerp

De lijst werkwoorden die de speler kan gebruiken staat in DATA-statements (beginnende

op regel 7001 in dit programma). Steeds als de speler een werkwoord gebruikt wordt deze lijst afgezocht om te kijken of het woord er in staat. Elk werkwoord heeft ook een nummer, gelijk aan de positie in de lijst, wat het programma gebruikt om de subroutine te vinden die de actie van het werkwoord regelt. Deze subroutine staat op 20 * woordnummer + constante (1480 in dit spel, de subroutine voor het eerste werkwoord staat dus op regel 1500, voor het tweede woord op 1520 enz.).

Programma-opbouw

Regel	Beschrijving
5 - 40	initialisatie variabelen met vaste waarden, schermparameters en speciale variabelen voor dit avontuur
50 - 110	besturingslus voor dit programma, bestaat uit: 1. zet locatiebeschrijving op scherm zet zichtbare voorwerpen op scherm 2. haal de opdracht van de speler op voer de opdracht uit als opdracht mislukt ga naar 2 ga naar 1
900 - 940	initialisatie speciale variabelen: het aantal acties en voorwerpen, de voorwerplocatie array om geheugen te besparen enkele strings die voor diverse locaties gelijk zijn enkele variabelen die als voorwaarden gebruikt worden, bijv. een deur die op slot zit
1000 - 1100	invoer opdracht speler. hoofdletters worden omgezet in kleine letters en de eerste twee ingetikte woorden komen in a$ en n$ (een woord is elke reeks karakters eindigend met een spatie) de werkwoordenlijst wordt afgezocht op a$ en als die gevonden wordt dan geeft de subroutine het woordnummer door aan het hoofdprogramma, zo niet dan wordt speler om een nieuwe opdracht gevraagd.
1200 - 1260	deze subroutine zoekt een bepaald woord op in bepaalde lijst, wordt gebruikt om de woorden voor een actie en de voorwerpbeschrijvingen bij een voorwerp op te zoeken
1300 - 1311	zoekt de voorwerpbeschrijvingen voor de n$ (het tweede door de speler ingetikte woord) als het zoeken resultaat op heeft geleverd dan worden beschrijving,

	voorwerpnummer en een vlag ten teken van resultaat opgegeven, zo niet dan wordt de vlag op geen resultaat gezet
1500 - 1802	deze serie subroutines worden gebruikt om de acties behorende bij de werkwoorden uit te voeren
1997	naar deze regel wordt gesprongen als een actie niet mogelijk is, de melding 'onmogelijk' verschijnt op het scherm en de vlag wordt op geen resultaat gezet
1998	als actie niet mogelijk is maar bovenstaande melding mag niet op scherm, vlag wordt wel gezet
1999	naar deze regel wordt gesprongen als actie mogelijk is, vlag wordt op resultaat gezet
2000 - 2050	geeft een lijst van alle voorwerpen met locatie gelijk aan variabele 1, indien deze er niet zijn verschijnt de melding 'niets'
7000 - 7004	bevat de werkwoordenlijst in regel 7000 staat het aantal werkwoorden
7100 - 7160	dit zijn de voorwaarden om voorwerpen op te kunnen pakken of locaties binnen te gaan. N.B. voorwaarde 0 wordt altijd aan voldaan, dus als er geen voorwaarden nodig zijn dan wordt voorwaarde 0 gegeven
8000 - 8525	data voor de voorwerpen
9001 - 9030	data voor de locaties
9980 - 9981	naar deze regel wordt na afloop van het spel gesprongen als speler alle schatten heeft gevonden
9990 - 9992	na afloop (als speler gewonnen heeft of vermoord is) wordt hier de score op het scherm gezet en wordt speler gevraagd of hij nog een keer wil spelen

Hoe moet u uw eigen avonturenspel opzetten?
Eerst heeft u een wereld nodig om te kunnen ontdekken. Dit betekent dat er een databestand opgebouwd moet worden van de beschrijvingen van alle locaties die bezocht kunnen worden en de verbindingen daartussen. Verder moet u besluiten welke voorwaarden er voor de locaties gelden.

Laten we als voorbeeld een wereld opzetten bestaande uit twee locaties:

1 een gang die doorloopt zover het oog reikt,
2 een kleine grot.

De gang ligt ten noorden van de grot en als u in de gang naar het zuiden gaat dan komt u in de grot. Andere richtingen zijn niet toegestaan. Als voorwaarde is er een deur tussen de twee locaties die op slot zit. De speler kan alleen van locatie verwisselen als de deur van het slot af is. U krijgt dus:

```
9001 DATA   "in een gang die doorloopt zover het oog reikt"
            0,2,0,0,0,0,0,
9002 DATA   "in een kleine grot",1,0,0,0,0,0,1
```

Merk op dat deuren maar een richting hebben er is dus geen voorwaarde verbonden aan het binnentreden van de gang, wel aan het binnengaan in de grot.
De gang is onze eerste locatie, hier begint de speler en hier moeten de schatten naar toe worden gebracht.
Nu heeft u wat voorwerpen nodig. U had al een afgesloten en een onafgesloten deur nodig (bij het begin van het spel staat de gesloten deur tussen gang en grot, maar als het slot verwijderd is dan stopt u de afgesloten deur in locatie -1 om daar vanaf te zijn en vervangt u door een onafgesloten deur). U heeft natuurlijk ook een sleutel nodig om de deur te openen. Als schat plaatst u een smaragd in de grot. Verder legt u een zwaard in de gang en laat u de smaragd door een slang bewaken.
U krijgt dus:

voorwerp	soort	kracht	voorwaarde	locatie
afgesloten deur	0	0	onmogelijk	1
open deur	0	0	onmogelijk	-1
sleutel	0	0	0	1
zwaard	0	10	0	1
slang	-1	100	onmogelijk	2
smaragd	30	0	2	2

Dit betekent dat de speler de deur en de slang niet kan oppakken en dat de smaragd alleen meegenomen kan worden als aan voorwaarde 2 is voldaan. Voorwaarde 2 wordt dan een test om te bepalen of de slang dood is.
Het zwaard heeft kracht 10 en de slang kracht 100, dus als de speler begint met kracht 100 en het zwaard heeft, dan zijn speler en slang aan elkaar gewaagd, met een licht voordeel voor de speler.
De smaragd geeft een score van 30, de slang is een monster en de rest zijn normale voorwerpen.

Dit wordt dan in het programma:

```
8000 DATA 6,1,-1,1,1,2,2
8001 DATA "afgesloten deur"
8002 DATA "onafgesloten deur"
8003 DATA "sleutel"
8004 DATA "zwaard"
8005 DATA "slang"
```

```
8006 DATA "smaragd"
8501 DATA 0,0,on-vo          (on is het adres van de
8502 DATA 0,0,on-vo          onmogelijke routine en vo
8503 DATA 0,0,0              is het voorwaarde-adres,
8504 DATA 0,10,0             on-vo is dus een voorwaarde
8505 DATA -1,100,on-vo       waar niet aan voldaan kan
8506 DATA 30,0,2             worden)
```

U heeft nu de acties nog nodig die uitgevoerd kunnen worden. De richtingen met bijbehorende subroutines en 'inventariseer', 'pak', 'val' en 'vermoord' zijn redelijk standaard en worden waarschijnlijk in al uw avonturen gebruikt. Kijk hiervoor in het programma. U heeft de actie 'open' voor de deur nodig, de werkwoordlijst wordt dan:

```
7000 DATA 11 (het aantal acties)
7001 DATA "noord","zuid","oost","west"
7002 DATA "hoog","laag","pak","val"
7003 DATA "vermoord","inventariseer","open"
```

U moet nu alleen de 'open'-routine nog schrijven. Deze moet testen of de speler in de gang is, de sleutel bij zich heeft, en of de deur nog op slot zit. U heeft een variabele nodig die aangeeft of de deur op slot zit (dit moet in het begin opgezet worden anders kan de deur open staan terwijl u denkt dat hij op slot zit). Deze routine ziet er als volgt uit:

```
1700 IF lu = 1 OR l <>  1 OR o(3) <>0
       THEN GOTO on     (testvoorwaarden)
1701 LET o(1) = -1      (verwijder gesloten deur)
1702 LET o(2) = 1       (andere deur in de plaats)
1703 LET lu   = 1       (zet variabele op 1)
1704 GOTO tr (succes)
```

Nu de voorwaarden nog. U heeft de volgende twee nodig:

1 is deur op slot,
2 is slang dood.

In het programma wordt dat:

```
7101 GOTO on+lu          (als lu=0 dan ga naar onmogelijk,
                         als lu=1 dan ga naar tr, succes, tr staat nl. op adres
                         on+1)
7102 GOTO on+ (o(5)=-1)
                         (als o(5)=-1 dan (o(5)--1)=1,
                         dus ga naar tr
                         (als o(5)<>-1 dan (o(5)=-1)=0,
                         dus ga naar on)
```

Nu heeft u dus een avonturenspel, hoewel een beetje aan de kleine kant. Veel geluk met het schrijven van uw eigen spel!!

Programma

```
   1 REM avontuur
   5 LET tr=1999: LET on=1998: LET fa=on: LET ls=9000: LET
os=8000: LET oz=8500: LET vs=7000: LET vo=7100: LET cs=vo:
LET sc=0
   6 LET vr=1480: LET z=0: LET ct=1997: LET dr=1601: LET cx
=6: LET st=100
  10 BORDER 5: PAPER 7: INK 0: CLS
  30 LET l=1
  40 GO SUB 900
  50 POKE 23692,255: RESTORE l+ls: READ m$: PRINT "Je#bent#
";m$: LET q$="Je#ziet": GO SUB 2000
  60 BEEP .2,12: PRINT
  70 GO SUB 1000
  80 PRINT v$;"#";
  90 GO SUB vr+vn*20
 100 IF f=0 THEN  GO TO 60
 110 PRINT : GO TO 50
 900 RESTORE os: READ oc: DIM o(oc)
 910 FOR i=1 TO oc: READ o(i): NEXT i
 920 RESTORE vs: READ vc
 930 LET be=0: LET lu=0: LET gu=0: LET e$="in#een#uitgestre
kte#####woesti jn": LET p$="in#een#nauewe#gang"
 940 RETURN
1000 INPUT 'WAT#MOET#IK#DOEN#';s$: IF LEN s$=0 THEN  GO TO
1000
1010 LET a$="": LET n$=""
1020 LET x=0: FOR i=1 TO LEN s$
1030 LET i$=s$(i TO i)
1040 IF i$>="A" AND i$<="Z" THEN  LET i$=CHR$ (32+CODE i$)
1050 IF i$="#" AND x=1 THEN  LET i=LEN s$: GO TO 1080
1051 IF i$="#" THEN  LET x=1: GO TO 1080
1060 IF x=0 THEN  LET a$=a$+i$
1070 IF x=1 THEN  LET n$=n$+i$
1080 NEXT i: LET w$=a$: LET c=vc: LET d=vs+1: GO SUB 1200
1090 IF vn=0 THEN  PRINT "INV IK#KEN#GEEN TRU #";a$: GO TO
1000
1100 LET v$=y$: RETURN
1200 LET vn=0
1210 RESTORE d
1220 FOR i=1 TO c: READ m$: LET x=LEN m$: IF x>LEN w$ THEN
 LET x=LEN w$
1240 IF w$( TO x)=m$( TO x) THEN  LET vn=i: LET y$=m$: LET
i=c
1260 NEXT i: RETURN
1300 IF n$="" THEN  GO TO 1311
1301 LET w$=n$: LET c=oc: LET d=os+1: GO SUB 1200: IF vn=0
THEN  GO TO 1311
1310 PRINT y$: GO TO tr
1311 PRINT "INV IK#KEN#GEEN TRU #";n$: GO TO fa
1500 LET d=1: GO TO dr
1520 LET d=2: GO TO dr
1540 LET d=3: GO TO dr
1560 LET d=4: GO TO dr
```

```
1580 LET d=5: GO TO dr
1600 LET d=6
1601 PRINT : RESTORE 1+1s: READ m$: FOR i=1 TO d: READ n1:
NEXT i
1602 IF n1=0 THEN  PRINT "JE#KUNT#DIE##KAN1#NIET#DOEN": GO TO
fa
1603 RESTORE n1+1s: READ m$,s,s,s,s,s,s
1604 READ s
1605 GO SUB cs+s: IF f=0 THEN  GO TO 1611
1610 IF o(1)=0 THEN  LET 1=n1: GO TO tr
1611 PRINT "IETS#HOUDT#JE#TEGEN": GO TO fa
1620 GO SUB 1300: IF f=0 THEN  RETURN
1621 IF o(vn)<>1 THEN  PRINT "DAT#1S#H1ER#NIET": GO TO fa
1622 RESTORE oz+vn: READ s: IF s<0 THEN  GO TO ct
1624 READ s
1625 READ s
1626 GO SUB cs+s: IF f=0 THEN  GO TO ct
1630 IF cx<>0 THEN  LET cx=cx-1: LET o(vn)=0: RETURN
1631 PRINT "JE#HEBT#TEVEEL#BIJ#JE": GO TO fa
1640 GO SUB 1300: IF f=0 THEN  RETURN
1641 IF o(vn)<>0 THEN  PRINT "DAT#HEB#JE#NIET#BIJ#JE": GO T
O fa
1642 RESTORE oz+vn: READ s
1643 LET o(vn)=1: IF 1=1 AND s<>0 THEN  LET sc=sc+s: LET o(
vn)=-1: PRINT "De#";y$;"#verdwijnt"
1644 LET cx=cx+1: IF sc=220 THEN  GO TO 9980
1645 RETURN
1660 GO SUB 1300: IF f=0 THEN  RETURN
1661 LET t$=y$: LET mn=vn: RESTORE oz+vn: READ s,ms: IF s>=
0 THEN  GO TO ct
1662 LET s=0: BEEP .1,20: INPUT "vermoord#waarmee#";w$: IF
 w$="" THEN  GO TO 1670
1663 LET c=oc: LET d=os+1: GO SUB 1200: IF vn=0 THEN  GO TO
 1662
1664 IF o(vn)<>0 THEN  GO TO 1662
1665 RESTORE oz+vn: READ s,s: IF s=0 THEN  GO TO 1662
1666 PRINT "#met#";y$;
1670 PRINT : LET s=s+st: IF ms>s+RND*20 THEN  GO TO 1679
1671 IF s>ms+RND*15 THEN  GO TO 1678
1672 PRINT "De#";t$;"#vecht#terug.#Je#wordt#zwakker"
1673 LET st=st-RND*5: GO TO tr
1678 PRINT "Je#hebt#de#";t$;"#vermoord": PRINT "Het#lijk#ve
rdwijnt#in#een#rookwolk": LET o(mn)=-1: GO TO tr
1679 PRINT "De#";t$;"#heeft#jou#vermoord": GO TO 9990
1680 LET 11=1: LET 1=0: LET q$="Je#hebt#bij#je": GO SUB 200
0: LET 1=11: GO TO tr
1700 PRINT sc: GO TO tr
1720 GO SUB 1300: IF f=0 THEN  RETURN
1721 IF vn<>20 OR o(2)<>0 THEN  GO TO ct
1722 LET o(14)=o(vn): LET o(vn)=-1: PRINT "Een#ivoren#sleut
el#komt#uit#het#plafond#gevallen": RETURN
1740 IF 1<>11 AND 1<>29 THEN  GO TO ct
1741 IF 1=11 AND o(3)<>0 THEN  PRINT "Je#zinkt#naar#de#bode
m#van#de###plas#en#verdrinkt": GO TO 9990
1742 IF 1=11 THEN  LET 1=29: GO TO tr
```

```
1743 LET l=11: GO TO tr
1760 GO SUB 1300: IF f=0 THEN  RETURN
1761 IF l=1 AND o(4)=0 THEN  LET o(22)=-1: LET o(23)=1: LE
 lu=1: RETURN
1762 IF l=13 AND o(14)=0 THEN  LET o(24)=-1: LET o(25)=1: L
ET gu=1: RETURN
1763 GO TO ct
1780 IF o(16)<>0 OR l<3 OR l>10 THEN  GO TO ct
1781 PRINT "Je#vindt#";
1782 IF i=6 AND o(4)=-1 THEN  LET o(4)=1: PRINT "iets": GO
TO tr
1783 PRINT "niets": GO TO tr
1800 GO SUB 1300: IF f=0 THEN  RETURN
1801 RESTORE oz+vn: READ s,s1,s1
1802 IF s<0 THEN  GO TO ct
1803 GO SUB cs+s1: IF f=0 THEN  GO TO ct
1804 PRINT "In#de#roos'": LET o(vn)=-1: RETURN
1820 GO SUB 1300: IF f=0 THEN  RETURN
1821 IF l<>24 OR vn<>7 THEN  GO TO 1830
1822 PRINT "Een#brug#verschijnt#over#de#####kloof": LET o(1
5)=1: LET be=1: GO TO tr
1830 PRINT "Er#gebeurt...niets": GO TO fa
1840 IF l=24 OR l=26 THEN  PRINT "De#val#heeft#je#nek#gebro
ken": GO TO 9990
1841 GO TO 1830
1860 GO SUB 1300: IF f=0 THEN  RETURN
1861 IF vn<>18 THEN  GO TO ct
1862 PRINT "Dat#was#heerlijk": LET o(19)=o(vn): LET o(vn)=-
1: RETURN
1997 PRINT "ONMOOEL IJK"
1998 LET f=0: RETURN
1999 LET f=1: RETURN
2000 LET x=0: PRINT q$
2010 RESTORE os+1: FOR i=1 TO oc: READ m$
2020 IF l<>o(i) THEN  GO TO 2030
2021 LET x=1: PRINT "##een";
2022 LET z$=m$(1 TO 1): IF z$="a" OR z$="e' OR z$="i" OR z$
="o" OR z$="u' THEN  PRINT "n";
2024 PRINT "#";m$
2030 NEXT i
2040 IF x=0 THEN  PRINT "##niets'
2050 RETURN
7000 DATA 19
7001 DATA "noord","zuid","oost','west","hoog","laag"
7002 DATA "pak",'val","vermoord","inventariseer",'score"
7003 DATA "breek","zwem","open','graaf"
7004 DATA "eet",'zwaai","spring","drink"
7100 GO TO tr
7101 GO TO fa+lu
7102 GO TO fa+gu
7103 GO TO fa+be
7104 GO TO fa+(o(3)=0)
7105 GO TO fa+(o(11)=-1)
7106 GO TO fa+(o(12)=-1)
7107 GO TO fa+(o(10)=-1)
```

```
7150 IF o(18)<>0 THEN  GO TO tr
7151 LET o(18)=-1: LET o(12)=-1: PRINT "Als#je#binnenkomt#s
teelt#een####piraat#je#rum#en#loopt#lachend##weg": GO TO tr

7160 PRINT "De#duivel#vermoordt#je": GO TO 9990
8000 DATA 25,1,12,27,-1,14,17,21,25,29,12,14,17,20,-1,-1,11
,2,2,-1,19,11,1,-1,13,-1
8001 DATA "koperen#lamp"
8002 DATA "degen"
8003 DATA "snorfel"
8004 DATA "grote#sleutel"
8005 DATA "persische#zijde"
8006 DATA "gouden#munt"
8007 DATA "zilveren#muur"
8008 DATA "robijn"
8009 DATA "diamant"
8010 DATA "gemene#trol"
8011 DATA "groene#draak"
8012 DATA "piraat"
8013 DATA "duivel"
8014 DATA "ivoren#sleutel"
8015 DATA "kristallen#brug"
8016 DATA "schop"
8017 DATA "keuken#tafel"
8018 DATA "fles#rum"
8019 DATA "lege#fles"
8020 DATA "touwladder#tussen#vloer#en#plafond"
8021 DATA "kleine#waterplas"
8022 DATA "gesloten#deur"
8023 DATA "deur"
8024 DATA "gesloten#poort"
8025 DATA "poort"
8501 DATA z,z,z
8502 DATA z,20,7
8503 DATA z,z,z
8504 DATA z,z,z
8505 DATA 10,z,5
8506 DATA 50,z,6
8507 DATA 20,z,z
8508 DATA 30,z,z
8509 DATA 100,z,z
8510 DATA -1,87,z
8511 DATA -1,110,z
8512 DATA -1,200,z
8513 DATA -1,200,z
8514 DATA 10,z,z
8515 DATA z,z,z
8516 DATA z,z,z
8517 DATA z,z,fa-cs
8518 DATA z,z,z
8519 DATA z,z,z
8520 DATA z,z,fa-cs
8521 DATA z,z,fa-cs
8522 DATA z,z,fa-cs
8523 DATA z,z,fa-cs
```

```
8524 DATA z,z,fa-cs
8525 DATA z,z,fa-cs
9001 DATA 'in#de#woonkamer#van#een#groot#huis.#Op#een#bord#
staat:##'Breng#alle#schatten#hier#terug'",2,z,z,z,z,12,z
9002 DATA 'in#de#keuken",z,1,3,z,z,z,z,z
9003 DATA e$,z,4,5,2,z,z,z
9004 DATA e$,3,7,6,z,z,z,z
9005 DATA e$,z,6,7,3,z,z,z
9006 DATA e$,5,8,z,4,z,z,z
9007 DATA e$,z,z,z,4,z,z,z
9008 DATA e$,6,9,z,z,z,z,z
9009 DATA e$,8,z,10,z,z,z,z
9010 DATA e$,11,z,z,9,z,z,z
9011 DATA 'bij#een#oase',z,10,z,z,z,z,z
9012 DATA "in#een#kelder",z,16,12,z,1,z,1
9013 DATA 'bij#de# EXT SHI 2 POORT#VAN#DE#HEL EXT SHI 0',
z,z,z,12,z,20,z
9014 DATA "in#een# INV aardsdonkere#grot TRU",z,15,z,z,z,z,
z
9015 DATA p$,14,10,z,16,z,z,z
9016 DATA p$,12,z,15,z,z,17,z
9017 DATA 'in#het#piratennest',z,z,z,z,16,z,50
9018 DATA p$,15,19,z,z,z,z,z
9019 DATA p$,18,z,z,z,z,z,z
9020 DATA "in#de#HEL...#Een#duivel#zegt#'Vind#de#uitgang#-#
 EXT SHI 2 of#sterf EXT SHI 0'",30,30,30,21,30,30,2
9021 DATA p$,z,24,22,20,z,z,z
9022 DATA "in#een#doodlopend#stuk",z,z,z,21,z,z,z
9023 DATA "EXT 0 EXT SHI 2DOOD#IN#DE#HEL EXT 7 EXT SHI 0",z
,z,z,z,z,z,z
9024 DATA 'op#de#rand#van#een#diepe#kloof",21,25,z,z,z,z,z
9025 DATA "in#een#schitterende#edelstenen##hal",24,z,27,z,z
,z,3
9026 DATA 'op#de#rand#van#een#diepe#schachtEr#zijn#hier#spo
ren#van# EXT SHI 2 vuur EXT SHI 0 #en# EXT 6 zwavel EXT 7",
z,27,z,z,z,23,z
9027 DATA p$,26,z,z,25,z,28,z
9028 DATA p$,z,z,z,z,z,27,18,z
9029 DATA "EXT SHI 1 aan#het#zwemmen#in#een#kleine#waterpla
s EXT SHI 0",z,z,z,z,z,z,z,4
9030 DATA "",z,z,z,z,z,z,z,60
9980 PRINT : PRINT : PRINT : PRINT
9981 PRINT "GEFELICITEERD##Het#avontuur#is##ten#einde#en#je
#bent#veilig#met#alle#schatten#thuisgekomen."
9990 PRINT : PRINT "SCORE#";sc: PRINT "Wil#je#nog#een#keer#
spelen#?"
9991 LET m$=INKEY$: IF m$="" THEN  GO TO 9991
9992 IF m$( TO 1)="y" THEN  GO TO 5
```

Enkele tips voor het spelen van avontuur

(1) Verdwaald in de woestijn?

Wanhoop niet. De woestijn is niet eindeloos. Laat wat voorwerpen achter op strategische plaatsen zodat u een kaart van de woestijn kunt maken, die heeft u nodig om de oase te vinden.

(2) U krijgt de deur van de woonkamer niet open?
Daar heeft u de sleutel voor nodig die ergens in de woestijn begraven ligt. Om die sleutel uit te graven heeft u weer een schop nodig, en die kunt u ergens buiten het huis vinden.

(3) Zit u vast in de HEL?
Doe wat de duivel zegt.

(4) U komt niet voorbij de piraat?
Probeer de rum.

(5) U kunt de draak niet vermoorden?
Blijf proberen.

(6) U kunt de sleutel van de poort niet vinden?
Probeer de touwladder.

(7) U kunt niet naar de overkant van de kloof?
Misschien zou een tovermuur wel uitkomst brengen.

(8) U kunt de laatste schat niet vinden?
Ga een eindje zwemmen.

Voor de bezitters van een ZX Spectrum is dit boek een bron van inspiratie bij het programmeren van hun computer
De BASIC-programma's in dit boek zijn zó uiteenlopend van karakter dat het onmogelijk is om een grootste gemene deler aan te geven
Voor de liefhebbers van spelletjes zijn er voldoende programma's te vinden om voorlopig aan het scherm en het toetsenbord gekluisterd te zijn
Zij vinden onder andere gokspellen als Blackjack en Fruitmachine en grafische spellen als het bekende Invaders.
Wil men de computer voor zakelijke doeleinden gebruiken, dan grijpt men al gauw naar programma's als Loonadministratie of Verkoopanalyse
Het is overigens aan de gebruiker om het schaakprogramma in dit boek ergens bij in te delen. Om dit onder spelletjes te rangschikken lijkt enigszins oneerbiedig.
Bij elk programma is een uitgebreide beschrijving van het principe, de opbouw en de werking opgenomen, zodat een goed inzicht wordt verkregen in het programmeren en een basis wordt gelegd voor het maken van een eigen programma.

ISBN 90 201 1718 1